Pattern Recognition 78 (2018) 291–306

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A database of unconstrained Vietnamese online handwriting and recognition experiments by recurrent neural networks

Hung Tuan Nguyen [a], Cuong Tuan Nguyen [a], Pham The Bao [b], Masaki Nakagawa [a,*]

[a] Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology, 2-24-16 Naka-cho, Koganei-shi, Tokyo, 184-8588 Japan
[b] Department of Mathematics and Computer Science, Ho Chi Minh City University of Science, 227 Nguyen Van Cu Street, District 5, Ho Chi Minh City, Viet Nam

ARTICLE INFO

Article history:
Received 12 February 2017
Revised 10 January 2018
Accepted 14 January 2018
Available online 2 February 2018

Keywords:
Vietnamese online handwriting
Database
Handwriting recognition
Recurrent neural network
Long short term memory

ABSTRACT

We present our efforts to create a database of unconstrained Vietnamese online handwritten text sampled from pen-based devices. The database stores handwritten text for paragraphs, lines, words, and characters, with the ground truth associated with every paragraph and line. We show a detailed statistical analysis of the handwritten text in this database and describe recognition experiments using several recent methods including the Bidirectional Long Short-Term Memory (BLSTM) network. Overall, our database contains over 480,000 strokes from more than 380,000 characters, which, at present, is the largest database of Vietnamese online handwritten text. Although Vietnamese script is based on a fixed set of alphabet letters, the recognition of Vietnamese online handwritten text poses a difficult challenge because of many diacritical marks, which usually result in delayed strokes during writing. We designed and implemented an online handwriting-collection tool to gather data, as well as a line-segmentation tool and a delayed-stroke-detection tool to analyze collected handwritten text. We also conducted a statistical analysis based on the writer profiles. We applied a number of the state-of-the-art recognition methods on unconstrained Vietnamese handwriting to evaluate their performance, including the BLSTM network, which is an efficient architecture derived from the Recurrent Neural Network (RNN) and is often applied to sequence labeling problems. The BLSTM network achieved 90% character recognition accuracy, despite many long sequences with several delayed strokes. Our database is allowed open access for research to stimulate the development of handwriting research technology.

1. Introduction

Recent years have seen a dramatic growth of pen-based and touch-based devices for both personal and business users, which has resulted in a large number of applications of these devices in education, business, and so on. At the same time, the demand for processing so-called digital ink and its online recognition has been increasing so that people can use these applications effectively. Hence, the research on online handwriting recognition is being revived and extended. Another motivating factor for continuing this research is that handwriting is still a natural, effective and convenient way for people to represent, keep and exchange information even while interacting with or via a computer.

Here, we define digital ink as a time sequence of strokes, and a stroke as a time sequence of pen/touch points sampled from pen/touch down point to pen/touch up point. We define an off-stroke as a vector from pen/touch up point to pen/touch down point. Hereafter, we use "pen" instead of "pen/touch".

In handwriting recognition research, databases of handwritten text play an important role, as they are useful for training and evaluating different recognition algorithms. Hence, many databases have been compiled in online recognition communities such as UNIPEN [1], IRONOFF [2], IAM-OnDB [3] and IBM-UB-1 [4] for Latin script, Kuchibue and Nakayosi [5] for Japanese, AltecOnDB for Arabic [6], CASIA, SCUT and ICDAR2013 for Chinese [7–9], CROHME for mathematical expressions [10] and so on. Consequently, there are many research publications on online handwriting recognition for Western, Japanese, Chinese, Arabic, Indian Scripts and mathematical expressions [11–18]. However, there is no available database for Vietnamese online handwriting; only a few offline databases of Vietnamese printed documents and scanned handwritten documents [19–21] are there. Vietnamese online handwriting database is not only essential for Vietnamese online handwriting recognition research but it would also be useful for online handwriting recognition in general.

* Corresponding author.
E-mail addresses: ntuanhung@gmail.com (H.T. Nguyen), ntcuong2103@gmail.com (C.T. Nguyen), ptbao@hcmus.edu.vn (P.T. Bao), nakagawa@cc.tuat.ac.jp (M. Nakagawa).

https://doi.org/10.1016/j.patcog.2018.01.013
0031-3203/

(a) A delayed stroke is caused by a diacritical mark.

(b) A delayed stroke is caused by a non-diacritical mark.

(c) A diacritical mark is not written as a delayed stroke.

Fig. 1. The relation between delayed strokes and diacritical marks.

Table 1
Characters which may cause delayed strokes in different languages.

Languages	Lowercase characters that may cause delayed strokes
Alphabet (4)	i,j,t,f
Nynorsk Norwegian (13)	à,é,ê,i,j,ò,ô,ó,ü,ö,ä,t,f
Portuguese (16)	á,à,â,ã,é,ê,i,í,j,ó,ô,õ,ú,ü,t,f
French (17)	â,à,é,è,ê,ë,i,î,ï,j,ñ,ô,ù,û,ÿ,t,f
Vietnamese (71)	á,à,ả,ã,ạ,â,ấ,ầ,ẩ,ẫ,ậ,ă,ắ,ằ,ẳ,ẵ,ặ i,í,ì,ỉ,ĩ,ị,ú,ù,ủ,ũ,ụ,ư,ứ,ừ,ử,ữ,ự ó,ò,ỏ,õ,ọ,ô,ố,ồ,ổ,ỗ,ộ,ơ,ớ,ờ,ở,ỡ,ợ é,è,ẻ,ẽ,ẹ,ê,ế,ề,ể,ễ,ệ, ý,ỳ,ỷ,ỹ,ỵ,đ,j,t,f

Diacritical marks (DMs) or diacritics are peculiar characteristics of Vietnamese presenting a challenge for online handwriting recognition. In Vietnamese, DMs along with 6 original vowels (a, i, u, e, o, y) are combined to create 66 different derivative vowels. These DMs are written above and below a vowel. DMs and some consonants letters of Đ, đ, j, t, f are often written with delayed strokes, where a delayed stroke for a character is defined to be a stroke written after writing some strokes for subsequent characters. Short-distance delayed strokes are written after only a few strokes, while long-distance delayed strokes are written after more than a few strokes, say after a sentence. We should emphasize that delayed strokes are not always written for diacritical marks, and diacritical marks are not always written as delayed strokes. Fig. 1(a) shows a common case in which a diacritical mark (stroke number 3) is written as a delayed stroke. Fig. 1(b) shows an example where a non-diacritical mark stroke is written as a delayed stroke, where the small bar stroke of ‘t’ is written as a delayed stroke. Fig. 1(c) presents a case where a diacritical mark (stroke number 4) is not written as a delayed stroke. As shown by these three different examples of handwritten text for the same word, diacritical marks are often, but not necessarily, written as delayed strokes and other strokes are sometimes written as delayed strokes depending on the writer and the situation.

The number of characters that may cause delayed strokes in Vietnamese is much larger than in Portuguese, French and other European languages, it is the largest among all the languages shown in Table 1.

In general, delayed strokes are one of the most difficult problems in online handwriting recognition due to the variations of stroke orders. For instance, in English handwriting, people may move the pen back to add the tittles above letter "i" or "j" or the small crossbar of letter "t" after writing the word that includes them, or add them after finishing the whole sentence. It depends on the writer’s style and the situation. According to [22], these DMs are usually eliminated in the preprocessing step since the recognition performance for English is still good without them. However, we cannot ignore these DMs, although they are small, for recognition in Vietnamese and other such languages because they are crucial in determining the pronunciation and the meaning of words.

Our objective is not only to build a Vietnamese online handwriting database but also to solve the online handwritten Vietnamese text recognition problem, which is related to solving the sequence labeling problem with long stroke sequences. If we are able to solve the delayed stroke problem with DMs in the Vietnamese script, the recognition approach can also be applied to other languages where DMs or delayed strokes are frequent.

Based on our previous experiences with online handwriting recognition systems, we identified the need to create an online handwriting database for Vietnamese as an essential step for conducting research on Vietnamese handwriting recognition. This database must consist of natural and unconstrained handwritten text. Moreover, as reported in [23], a large number of training samples are needed for training classification algorithms. Consequently, we decided to build a large Vietnamese online handwriting database named HANDS-VNOnDB (VNOnDB in short). We started to collect unconstrained Vietnamese online handwritten text from Vietnamese people from September 2014.

Our earlier paper reported the initial collection of handwritten text from 80 Vietnamese people and presented preliminary experiments on the database using simple online features and the BLSTM network [24]. This current paper describes the full database collected from 200 people, shows a detailed statistical analysis, especially on delayed strokes, and presents recognition experiments using the architectures of RNN, Long Short-Term Memory (LSTM) and BLSTM employing more features. Besides the ground truth at the paragraph level (used in the earlier paper), the ground truth at the line level is also prepared and used to demonstrate the ability of the BLSTM network to recognize characters with short-distance delayed strokes.

The rest of this paper is organized as follows. Section 2 describes the requirements for VNOnDB and its design principles with related tasks. Section 3 presents the corpus and the ground-truth text preparation. Section 4 shows the pattern collection and verification processes. Section 5 presents an analysis of VNOnDB to clarify its characteristics. Section 6 gives more details of the analysis of delayed strokes. Section 7 reports our experiments on unconstrained Vietnamese online handwriting recognition based on the state-of-the-art methods, especially BLSTM. Finally, the conclusions are presented in Section 8.

2. Design principles

The purpose of VNOnDB is to provide a benchmark of recognition methods and algorithms for Vietnamese Online Handwriting recognition. Since these methods and algorithms are used on smartphones and tablets, the handwritten text should be collected as written by people on these devices.

This section presents the requirements for VNOnDB and presents the design principles to meet the requirements while making compromises with time and cost restrictions. Then, we present some tasks for presenting VNOnDB as a benchmark.

2.1. Requirements for VNOnDB

We propose the following requirements for creating a database of unconstrained online handwritten text (VNOnDB):

R1: Online handwritten Vietnamese text should be collected from Vietnamese people.

R2: All the commonly used Vietnamese words and symbols should be collected in order to obtain a good representation of Vietnamese handwritten text. A considerable amount of handwritten text should be collected from each person, so that VNOnDB will be useful for writer adaptation or identification.

R3: Handwritten scripts should be collected from as many people as possible, since handwriting includes individual variations and deformations, so that writer-independent recognition can be tested.

R4: Multiple lines of handwritten text should be collected since people can write them on expanding writing surfaces of tablets or pen-based computers. Collected patterns should be useful to evaluate the line and character segmentation as well as handwritten text recognition.

R5: Handwritten text should be collected with meaningful text because in such situations people write naturally and casually. Moreover, handwriting recognition is associated with linguistic processing, so language models should also be evaluated.

2.2. Design principles of VNOnDB

In order to satisfy the requirements while achieving compromises with time and cost restrictions, we set up the following design principles.

First, we collect handwritten text using ground-truth text. To collect handwritten text with natural deformations and variations, asking participants to write anything they like may seem to be the best, but this approach takes time for the participant to think what to write and also requires large effort in labeling. Moreover, collected patterns may not cover the range of words as expected. To avoid these problems, the ground-truth text was prepared from a corpus and people were asked to write it in English [3] and in Japanese [5]. In this method, the ground-truth text is automatically associated with handwriting and necessary patterns are collected. Although the handwritten text in this situation is not completely natural, people tend to show natural deformations and variations when they write meaningful text. This satisfies the requirements R2, R5 and R1 partially.

Secondly, due to the enlarged writing surfaces and the requirement to collect natural and meaningful patterns, we let the participants write paragraphs in multiple lines. This satisfies the requirements R1, R4 and R5.

Thirdly, in order to meet the purpose of VNOnDB, we also prepared a collection tool on a pen-based computer, which presents ground-truth text page by page and records online handwritten text. The collection tool is useful since we could give it to our collaborators in Vietnam for collecting handwritten texts from many people. The tool is helpful for meeting the requirement R3.

Fourthly, we labeled handwritten text both at the paragraph level and at the line level, which satisfies the requirement R4. Since we allowed people to write a paragraph with arbitrarily starting a new line, we applied text-line segmentation on the collected handwritten text for obtaining the line-labeled handwritten text. We refer to the original VNOnDB with ground truth being tagged at the paragraph level as VNOnDB-Paragraph and call the VNOnDB with ground truth at the line level as VNOnDB-Line. Character level ground-truth is left for future research since it is time consuming and VNOnDB is still useful as a benchmark.

Finally, we collected patterns from 200 people following the number of writers in the previous databases [3–5]. In addition, all participants were selected randomly by our collaborators in Vietnam, thus, there are variations in age, gender, and occupation of the participants. This satisfies R3.

2.3. Related tasks for benchmarking

In this section, we describe some tasks for presenting VNOnDB as a benchmark.

2.3.1. Text-line segmentation

Text-line segmentation is applied to separate text lines, provide ground truth for each segmented text line and validate it manually. In general, online handwritten text contains temporal information of pen trajectory and off-strokes, which can be employed for segmentation.

2.3.2. Delayed strokes detection and annotation

Secondly, we detect delayed strokes and annotate them since VNOnDB is used to evaluate the text segmentation and recognition performance on handwritten text with delayed strokes. A heuristic method, based on the position of current and the previous strokes on the same line, is applied.

2.3.3. Recognition experiments by the state-of-the-art methods

Finally, in order to confirm the criticality of VNOnDB, the state-of-the-art recognition methods RNN, LSTM, BLSTM networks combined with the Connectionist Temporal Classification (CTC) are applied to recognize the handwritten text in VNOnDB. The features for online handwriting recognition problem are extracted according to [22].

3. Ground truth preparation

3.1. Vietnamese language

The Vietnamese language has been documented as an Austroasiatic language that has been using Latin alphabet accompanying the diacritical marks since the 20th century. According to a contemporary printed Vietnamese dictionary [25], there are approximate 40,000 entry words containing 2000 to 3000 popular words. Table 2 shows the entire Vietnamese character set of 216 characters including 93 lower-case letters, 93 upper-case letters, 10 digits, 19 symbols and ‘space’. There are 72 vowels including 6 basic vowels and 66 vowels derived from them using DMs. Moreover, it contains digits and symbols as shown in Table 2. Four alphabet characters “f, j, w, z” were not there originally but are now used in foreign names.

3.2. Ground truth preparation

A reasonable ground-truth text should contain all of the above characters occurring in common sentences and include modern Vietnamese words from reliable resources. For this, we employ the

Table 2
Vietnamese character set. (Characters appearing rarely are shown in gray).

Upper-case vowels with DM	Á À Ả Ã Ạ Â Ấ Ầ Ẩ Ẫ Ậ Ă Ắ Ằ Ẳ Ẵ Ặ
	É È Ẻ Ẽ Ẹ Ê Ế Ề Ể Ễ Ệ
	Í Ì Ỉ Ĩ Ị
	Ó Ò Ỏ Õ Ọ Ô Ố Ồ Ổ Ỗ Ộ Ơ Ớ Ờ Ở Ỡ Ợ
	Ú Ù Ủ Ũ Ụ Ư Ứ Ừ Ử Ữ Ự
	Ý Ỳ Ỷ Ỹ Ỵ
Upper-case letters without DM	A B C D Đ E G H I K L M N O P Q R S T U V X Y
	F J W Z
Lower-case vowels with DM	á à ả ã ạ â ấ ầ ẩ ẫ ậ ă ắ ằ ẳ ẵ ặ
	í ì ỉ ĩ ị
	ú ù ủ ũ ụ ư ứ ừ ử ữ ự
	é è ẻ ẽ ẹ ê ế ề ể ễ ệ
	ó ò ỏ õ ọ ô ố ồ ổ ỗ ộ ơ ớ ờ ở ỡ ợ
	ý ỳ ỷ ỹ ỵ
Lower-case letters without DM	a b c d đ e g h i k l m n o p q r s t u v x y
	f j w z
Digits	0 1 2 3 4 5 6 7 8 9
Symbols	? / * - + ! , . " ' : ; # % & ( ) [ ]

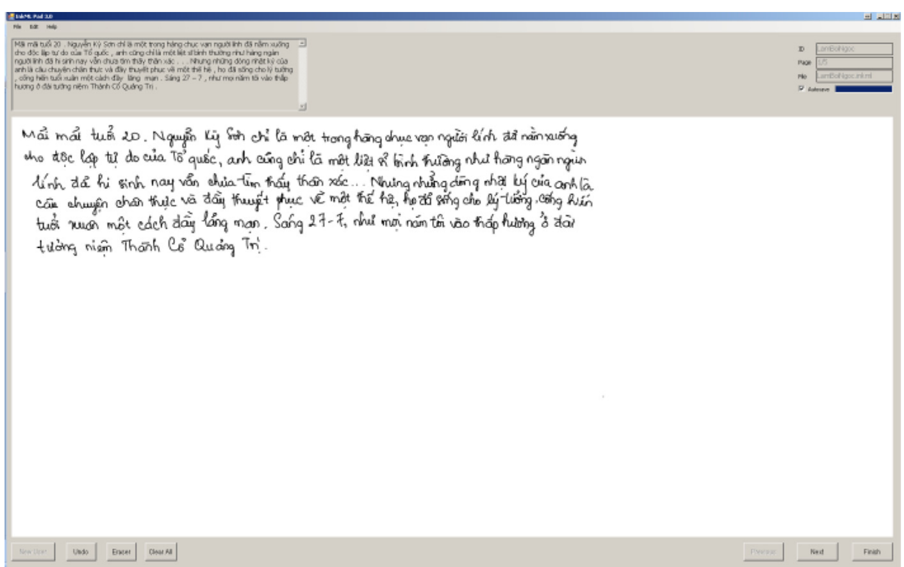


Fig. 2. Collected patterns according to the ground-truth text from a participant using our collection tool.

Vietnamese Treebank corpus [26] that includes about 10,000 annotated sentences by the grammar tags with approximately 300,000 characters. Sentences were extracted from the above corpus by eliminating all their grammar tags. Then, they were concatenated to make paragraphs, each of which can be written in a window on a tablet as shown in Fig. 2.

The corpus contains most of the upper-case and lower-case characters, but 57 characters (50 upper-case characters, 2 lower-case characters and 5 symbols), shown in gray in Table 2, are missing. We did not collect samples for the missing characters with the result that the corpus contains patterns for 216−57 = 159 characters. However, we include these 57 characters in the character set of VNOnDB so that it is possible to expand the database in the future.

4. Pattern collection and verification

4.1. Collection strategies

After preparing the ground-truth text, the following procedure was followed to collect natural handwritten text from the participants.

1. We explained the purpose of our research and requested the participants to write down their profile (gender, age, dominant hand, writing hand, occupation, and nationality).

2. We asked each participant to read the paragraph shown in the upper window, and write the text of the paragraph in the lower window as naturally as possible. This process is repeated for each page. Recognition results were not displayed. A large area

```
<ink>
  <definitions />
  <annotationXML>
    <Description>Cursive online handwriting</Description>
    <Content_Category>Text</Content_Category>
    <Language>Vietnamese</Language>
    <Writer_ID>xxxxxxxxxxx</Writer_ID>
    <Name>xxxxxxxxxxx</Name>
    <Gender>Female</Gender>
    <Age>21</Age>
    <Dominant_Hand>Right</Dominant_Hand>
    <Writing_Hand>Right</Writing_Hand>
    <Job>Student</Job>
    <Native_Language>
    </Native_Language>
    <Start_Time>2014-09-18T11:52:41</Start_Time>
    <DevName>FujitsuTabletPC</DevName>
    <SamplingRate>120</SamplingRate>
    <MaxNormalPressure>255</MaxNormalPressure>
    <Gt_File_Name>C:\\corpus_0001.txt</Gt_File_Name>
  </annotationXML>
  <traceGroup id="tg_0">
    <annotationXML>
      <Tg_Truth>Mãi mãi tuổi 20 . Nguyễn Kỳ Sơn chr là một trong hàng chục vạn người lính đã nằm
xuống cho độc lập tự do coa Tổ quốc , anh cũng chr là một liệt sĩ bình thường như hàng ngàn người
lính đã hi sinh nay vẫn chưa tìm thấy thân xác . . .
Nhưng những dòng nhật ký coa anh là câu chuyện chân thực và đầy thuyết phục về một thế hệ , họ đã
sống cho lý tưởng , cống hiến tuổi xuân một cách đầy  lãng  mạn . Sáng 27 - 7 , như mọi năm tôi
vào thắp hương o đài tưởng niệm Thành Cổ Quảng Trị .
</Tg_Truth>
    </annotationXML>
    <trace id="tr_0_0">673 857 ,673 857 ,659 794 ,659 794 ,659 794 ,650 734 ,650 734 ,650
734 ,650 734 ,650 734 ,650 734 ,650 734 ,650 734 ,650 734 ,650 734 ,668 800 ,668
800 ,674 865 ,674 865 ,682 935 ,682 935 ,694 1010 ,694
1010 ,700 1078 ,700 1078 ,707 1129 ,707 1129 ,707 1129 ,707 1129 ,707 1129 ,707 1129 ,707
1129 ,707 1129 ,707 1129 ,676 1071 ,676 1071 ,656 1011 ,656 1011 ,641 947 ,641 947 ,628 887
```


Fig. 3. An example of InkML file in VNOnDB.

on the handwritten text collection tool was provided, as shown in Fig. 2, to collect handwritten text as naturally as possible.

3. We did not use any box or guideline in the writing area for the participants as shown in Fig. 2, because our aim was to obtain handwritten text without any restriction. Hence, the sampled text could be curved or irregular. Due to the unconstrained handwritten text, layout analysis such as line segmentation becomes a challenging task.

4. To collect patterns on a large writing surface, we employed one of the most popular tablets, with a 74 mm × 184 mm handwriting area, and a sampling rate exceeding 120 Hz for obtaining high-resolution trajectories.

4.2. A tool for collecting handwritten text

Based on the procedure outlined in Section 4.1, a collection tool was designed and implemented for recording and storing the pen trajectory in the entire writing area via ink-space (HIMETRIC) by the unit of 0.01 mm, which is independent of the display resolution. Thus, the tool can be used on devices with different resolutions. This collection tool records not only the pen trajectory but also the stroke duration time and pressure information. Our partners in Vietnam were provided with this tool for collecting handwritten text patterns from their students and staff.

The Ink Markup Language (InkML) format [27] was used for storing the handwritten text since it is one of the most popular data formats to store and represent digital ink from tablets or pen-based computers. In VNOnDB, the handwritten texts of individual participants with particular ground-truth text files were stored in separate InkML files as shown in Fig. 3. The ground-truth labels for sequential ink data were assigned at the paragraph level. Therefore, we need to solve the unconstrained handwriting segmentation problem besides the whole paragraph recognition problem.

To protect the participant's personal information, we eliminated the writers' names out of InkML files before publishing the database, while retaining, in the specification part of each InkML file, the information about their age, gender, dominant hand, writing hand and occupation, as well as the technical information such as the device name, sampling rate, and the writing task.

4.3. Verifying handwritten text

The collected handwritten text from the writers need to be verified to avoid erroneous patterns that conflict with the ground truth because these erroneous patterns are meaningless and even harmful for training handwriting models. In order to point out erroneous patterns from the collected texts, we applied computer-assisted text verification and then employed manual verification to confirm whether they are indeed in error. When we made databases for Japanese characters in our previous work [5], we employed a character recognizer. This time, however, we employed measurements of pattern features, because we started this project before making a Vietnamese character recognizer. This method measured, for each character pattern, the average number of pen points, the average number and the average length of off-strokes. For each page by each writer, the average amount of pen points per character was determined. In addition, the off-strokes were grouped into three clusters: new-line off-strokes (between the last point of a previous line and the first point of a current line), long off-strokes (between two words) and short off-strokes (off strokes within a word). Then the average number and average length were computed for each cluster of off-strokes. Finally, the outliers according to the measurements, which are candidates for erroneous patterns, were extracted based on the K-nearest neighbor algorithm.

After the pattern verification, each writer confirmed the detected errors and wrote the missing text or rewrote the incorrect text (if he/she acknowledges the error). For three writers, however, we changed the ground-truth text to conform to their handwriting since they were not available.

5. Database analysis

Before doing any further research on VNOnDB, we analyze it from different perspectives to validate its suitability for handwriting recognition research.

5.1. Overall statistics

The latest version of VNOnDB stores handwritten texts from 200 people have the average age of 22.0 (with the standard deviation of 4.4). Fig. 4 shows the age distribution of the participants. Most collected handwritten texts are from adolescent participants, who are potential customers for smart devices. The oldest is 61 and the youngest 16, which covers a wide age range. Besides, the gender of participants is roughly balanced: 112 males (56.0%) and 88 females (44.0%). As most of the participants came from our col-

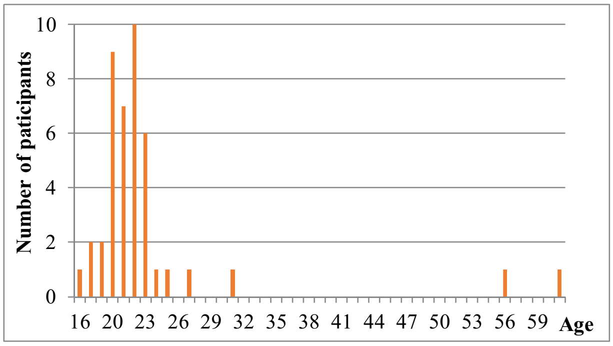


Fig. 4. Age distribution of writers.

Table 3
Statistics of VNOnDB.

Writers	Total	200
	Male & Female	Male 112 Female 88
	Age (from 16 to 61)	Average 22.0 Std.Var. 4.4
	Occupation	Student 196 and Staff 4
	Number of Characters in the character set	216
	Number of Files	255
	Number of Paragraphs	1146
	Number of Lines	7296
	Number of Words	111,641
	Number of Characters	386,326
	Number of Strokes	480,738

Table 4
Writing hand and Dominant hand.

Dominant hand\Writing hand	Left	Right
Left	1	8
Right	0	191

laborating university in Vietnam, 196 participants were students and 4 were faculty/staff.

Among all 216 characters shown in Table 2, 57 characters including the upper-case characters are not used often and they do not appear in the ground-truth text so that their handwritten character patterns are not stored in VNOnDB. According to Table 3, VNOnDB stores more than 480,000 strokes of over 380,000 characters (without the space character). Therefore, VNOnDB is the largest Vietnamese online handwriting database, which is comparable with databases in other languages. Table 4 shows that 191 people (95.5%) are right-handed and write with the right hand, 8 people (4%) are left-handed and write with the right hand and 1 person (0.5%) is left-handed and writes with the left hand. This is because all students in Vietnam must basically use the right hand for writing from primary school even if they are left-handed.

5.2. Statistics on handwritten text patterns

We analyzed further characteristics of handwritten text patterns in VNOnDB via distributions of the numbers of lines, words, characters and strokes. Note that the size of the writing surface is 74 mm height × 184 mm width. Fig. 5 shows the distribution of the number of lines to write a paragraph. It ranges from 1 to 12 lines with an average of 6.37 lines. Thus, most of the collected patterns in VNOnDB contain multiple lines.

Next, Fig. 6(a) shows the distribution of the number of words per line. It ranges from 1 to 35 words with an average of 15.30 words. In addition, the number of characters without space is between 2 and 124 per line, with the average being 52.95 characters, as shown in Fig. 6(b).

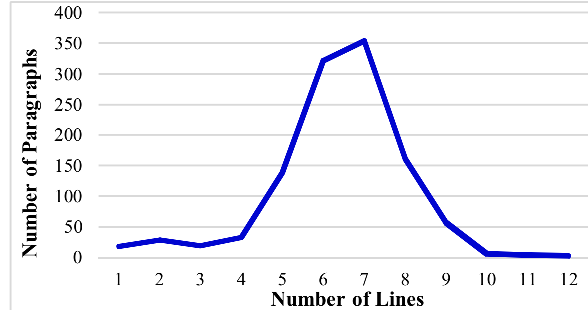
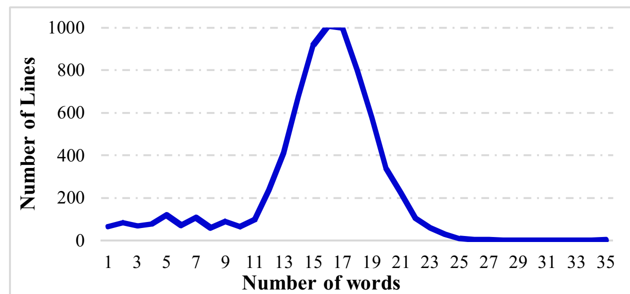
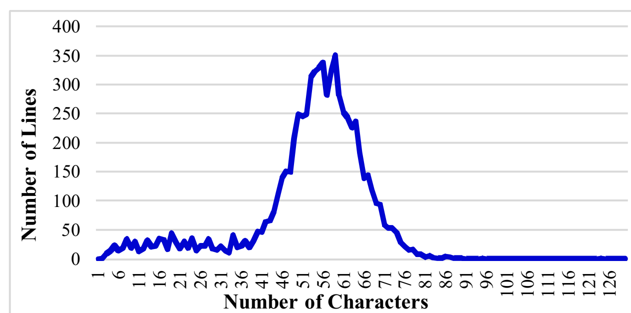


Fig. 5. Distribution of the number of lines in VNOnDB.



(a) Distribution of the number of words per line.



(b) Distribution of the number of characters per line.

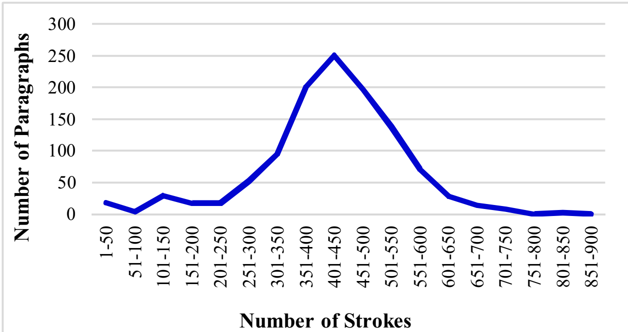
Fig. 6. Word and character distribution in VNOnDB.

We also analyzed the distribution of the number of strokes per paragraph and per line, as shown in Fig. 7(a) and (b). The number of strokes is between 12 and 820 per paragraph and between 1 and 155 per line. The average number of strokes is 420.23 per paragraph and 65.86 per line. Based on this analysis, the average length of stroke sequences is more than 400, which is challenging for transcribing whole paragraphs with multiple lines.

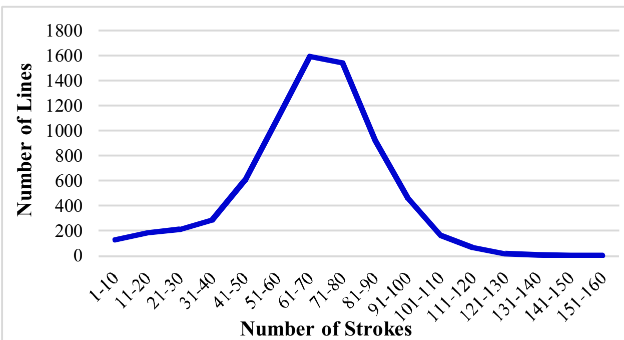
5.3. Varieties of handwritten text patterns in VNOnDB

Various styles of handwriting were collected: some styles are illustrated in Fig. 8. In Fig. 8(a), different people wrote the same words in various styles. As shown in Fig. 8(b), many strokes of these characters are connected, and sometimes multiple strokes are simplified to a single stroke. Even humans find it hard to recognize these characters. Fig. 8(c) shows some examples of such patterns in VNOnDB, which are erroneous and difficult to read without context. They often contain redundant or discontinuous strokes.

As mentioned in Section 4, we employed pattern verification process after pattern collection. In some cases, however, a writer did not rewrite his or her patterns because these patterns, though



(a) Distribution of the number of strokes per paragraph.



(b) Distribution of the number of strokes per line.

Fig. 7. Stroke distribution in VNOnDB.

erroneous, were still readable in the context. For handwriting recognition in practical environments, such patterns are challenging. Therefore, we keep these erroneous patterns in VNOnDB to evaluate the robustness of handwriting recognition algorithms.

Moreover, there are many different text lines for the same ground-truth text because participants wrote in multiple lines without any constraint.

6. Text-line segmentation and delayed strokes detection

6.1. Text-line segmentation

In order to analyze unconstrained patterns in VNOnDB, we applied text-line segmentation. In general, text-line segmentation for online handwritten text is easier than for offline handwritten text because for online patterns, temporal information of pen trajectory is available. This temporal information contains not only time-sequences of pen points but also off-strokes between strokes.

However, when applying temporal information and off-strokes for segmentation, the large number of delayed strokes in Vietnamese online handwritten text creates ambiguity. Therefore, instead of employing off-strokes directly for text-line segmentation, we employed the following method with two steps.

In the first step, we nominated newline candidates among off-strokes based on least-squares estimation. If the next stroke from the current off-stroke could not be combined with the previous strokes to form a line, the current off-stroke is considered as a candidate for a new-line off-stroke. In the second step, we employed the K-means method for 5 clusters using extracted features of beginning points, ending points, directions, and lengths from each off-stroke. These 5 clusters are supposed to represent new-line off-strokes, short off-strokes, long off-strokes, delayed off-strokes in the same line, and delayed off-strokes over lines. We need the cluster of new-line off-strokes, so we selected the cluster among the 5 clusters containing most of the off-strokes detected in the first step.

With this method, most of our multi-line patterns could be segmented line-by-line as shown in Fig. 9. Finally, we confirmed and corrected erroneous text-line segmentations visually and manually.

6.2. Delayed strokes detection

Text-line segmentation is useful for reducing our effort towards further analysis: to detect delayed strokes, to compute the average width of characters, to evaluate text line recognition and so on. Since all Vietnamese students learn to write characters in specified stroke orders – to write diacritical marks immediately after vowels and also to write a small bar or a dot immediately after main bodies of ‘t’, ‘đ’, ‘i’, ‘j’ — we consider them as the correct stroke order for writing Vietnamese text. Recall that in Section 1, we defined delayed strokes as strokes that people move the pen back to write. They are written in a different stroke order compared with the correct stroke order for the ground truth.

According to this definition, in order to compare the order between strokes and characters, we have to group all strokes into isolated characters. In Fig. 10, we need to group the 1st and a part of the 2nd stroke for the character ‘t’, the rest part of the 2nd stroke for the character ‘h’, the 3rd stroke for ‘a’ ‘n’ and ‘h’ . Next, we need to detect the diacritical mark ‘̀’ (the 4th stroke) as a delayed stroke as it was written after the other characters ‘n’ and ‘h’. Even after having performed the line segmentation and having the line-level ground truth, much effort is still required to obtain the character-level ground truth.

Thus, we see that if we follow this definition, it is difficult to detect delayed strokes automatically, because we cannot make the perfect correspondence between all input strokes and those in the ground truth without having access to the character-level ground truth. Therefore, we employ heuristics for rough estimation of delayed strokes i.e., we treat any stroke as delayed if at least one of the following conditions is satisfied:

Condition 1 ($C1$):

$$\frac{x\_displacement}{average\ width\ of\ characters} \leq \delta$$

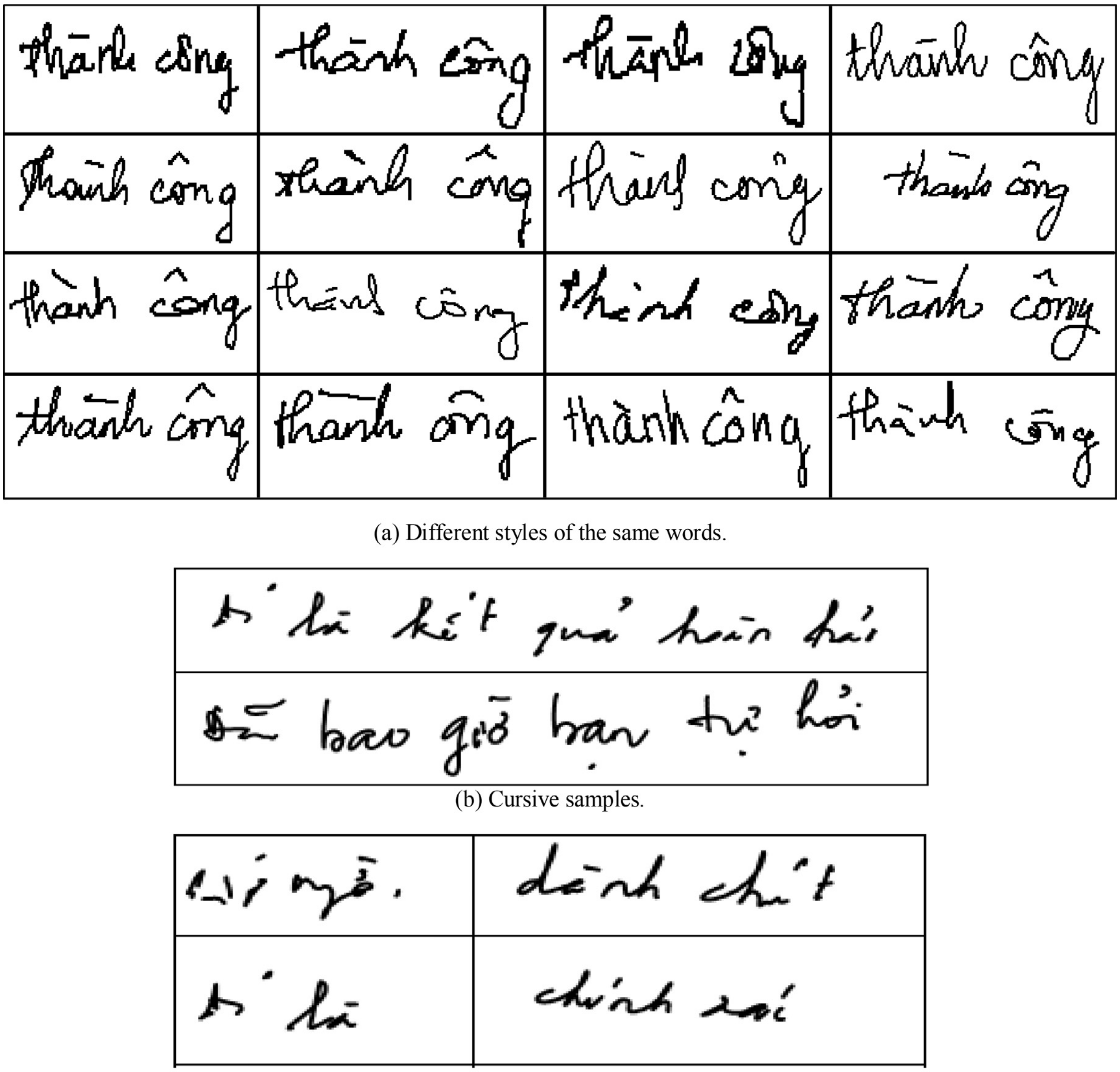
where ‘x_displacement’ is the horizontally projected displacement from the rightmost position of the previous strokes to the right side of the current stroke as shown in Fig. 11, and the average width of characters for each line is computed as the total length of non-zero horizontal projection in that line divided by the number of characters. We set the parameter δ as -1.0, which means that the current stroke is written in the negative direction by at least one character width.

Condition 2 ($C2$):

$order\ of\ the\ current\ stroke - order\ of\ the\ main\ stroke > 1$

where the ‘main stroke’ of the current stroke is defined as the stroke written before the current stroke and which overlaps with the current stroke on the x-coordinate. If the main stroke of the current stroke does not exist, the current stroke does not satisfy $C2$. On the other hand, if there are two or more previous strokes overlapping with the current stroke on the x-coordinate, we consider the stroke having the largest overlap with the current stroke as the main stroke. Condition $C2$ means that if the order of the current stroke minus that of the main stroke is greater than one, then the current stroke is a delayed stroke. Because the number of strokes to write Vietnamese characters is usually one or two and sometimes three, the stroke between the main stroke and the current stroke can be considered as a stroke for other characters.

Fig. 11 illustrates some delayed strokes and non-delayed strokes. The number next to each bounding box shows the writing order of the corresponding stroke. In each subfigure of Fig. 11, the



(a) Different styles of the same words.

(b) Cursive samples.

(c) Erroneous samples.

Fig. 8. Examples of pattern variations in VNOnDB.

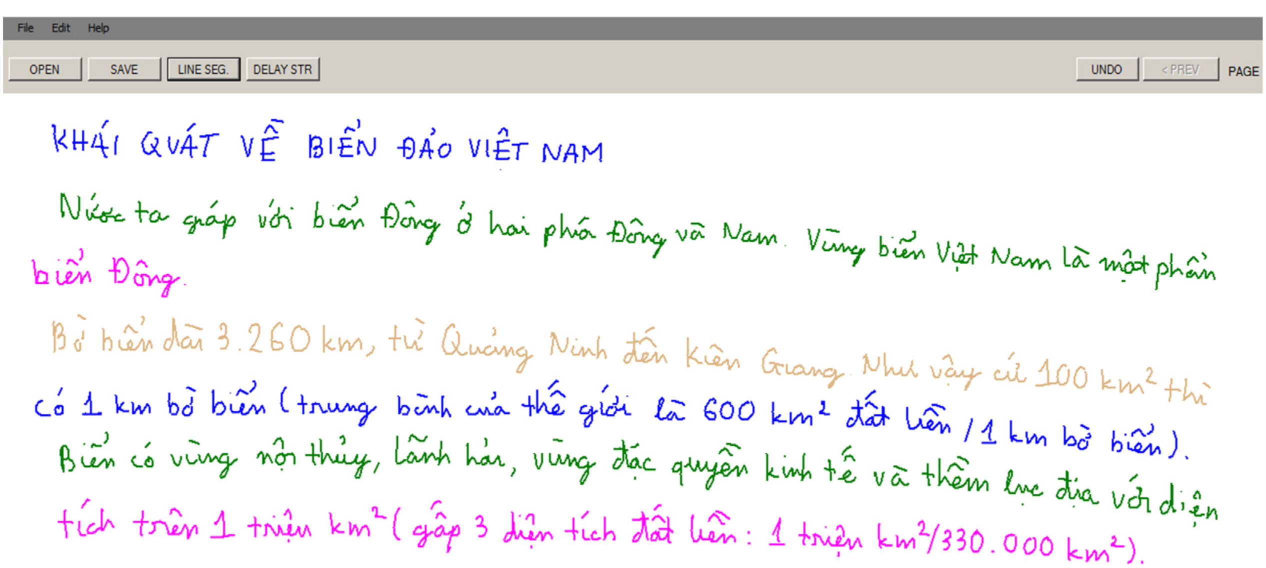
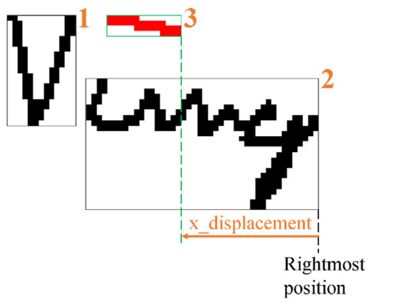
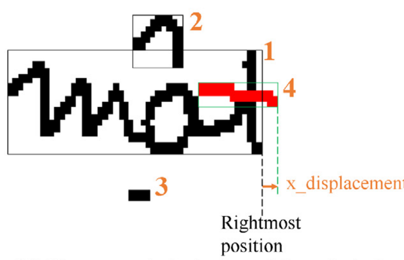


Fig. 9. An example of text-line segmentation in VNOnDB.

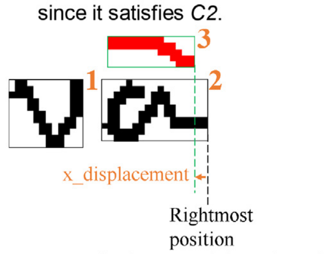
Fig. 10. The stroke order is "t h anh ‘ con ^ g" while the character order is "t h a ‘ n h c o ^ n g".



(a) The current stroke is a delayed stroke since it satisfies *C1*.



(b) The current stroke is a delayed stroke since it satisfies *C2*.



(c) The current stroke is not a delayed stroke since it does not satisfy any condition.

Fig. 11. Delayed strokes detection method.

current stroke is colored in red and the previous strokes are shown in black. Fig. 11(a) shows the current stroke with a large backward regression, which satisfies *C1*. Thus, the current stroke in Fig. 11(a) is a delayed stroke. Although the current stroke in Fig. 11(b) does not satisfy *C1*, it is still a delayed stroke because the current stroke is the 4th stroke and its main stroke is the 1st stroke, so it satisfies *C2*. Moreover, the 2nd and the 3rd strokes in Fig. 11(b) are also delayed strokes, because the 2nd stroke satisfies *C1* and the 3rd stroke satisfies both *C1* and *C2*. The current stroke in Fig. 11(c), however, is not a delayed stroke because it does not satisfy either *C1* or *C2*.

Fig. 12 shows that even though we apply simple heuristics, the delayed strokes are detected correctly, most of which are DMs. Moreover, we keep information on detected delayed strokes in separate files instead of adding them to InkML files.

6.3. Delayed strokes analysis

According to Table 1, all the 66 vowels, of both lower case and upper case, with DMs, and the letters Đ, đ, i, j, t, f may cause delayed strokes in online patterns. There are 138 such categories, which take more than 63% of all Vietnamese character categories. On the other hand, the number of character patterns having DMs in VNOnDB is about 38.89% of the whole text because almost all Vietnamese words are formed by a vowel and one or more consonants.

In term of strokes in VNOnDB, 23.70% of strokes are detected as delayed strokes by the method described in the previous section. Fig. 13 shows the distribution of the number of strokes separating the main stroke and the delayed stroke. Most of the delayed strokes are short-distance delayed strokes. Among 23.70% of the delayed strokes, 18.5% of strokes are short-distance delayed strokes while 5.2% are long-distance delayed strokes.

A stroke written right after the main stroke is considered as a delayed stroke if it satisfies either *C1* or *C2*. Such cases occur when a stroke is written after the main stroke and the subsequent strokes are cursively written as shown in Fig. 14. Some characters are cursively written such as "th", "anh", and "con" in Fig. 14(a), "anh" and "cong" in Fig. 14(b), and then DM's are written afterward. This causes many variations in writing delayed strokes. For example, Fig. 14 shows two different stroke orders in which different writers wrote the words "thành công".

7. Experiments and evaluations

7.1. Unconstrained handwriting recognition problem

We applied a number of state-of-the-art recognition methods to confirm the practicality of VNOnDB, which is a challenging target of the unconstrained sequence labeling problem with many delayed strokes. There are several efficient approaches for online handwriting recognition problem: for instance, Hidden Markov Models (HMMs) [28,29], Markov Random Fields (MRFs) [30,31], Conditional Random Fields (CRFs) [32,33], Recurrent Neural Networks (RNNs) [34] and more specifically Long Short-Term Memory Neural Networks (LSTMs) [35,36].

Among these approaches, the state models for HMMs or the features for CRFs need the prior knowledge and assumptions with respect to pattern dependencies. On the other hand, RNNs do not need any prior knowledge or assumption on pattern dependencies for the sequence-labeling problem. Moreover, the gradient descent algorithm is employed for training RNNs, which results in

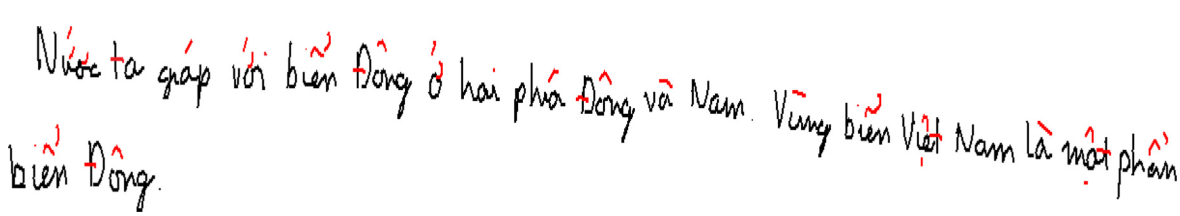


Fig. 12. Delayed strokes are detected and colorized in red color among which most of them are DMs.

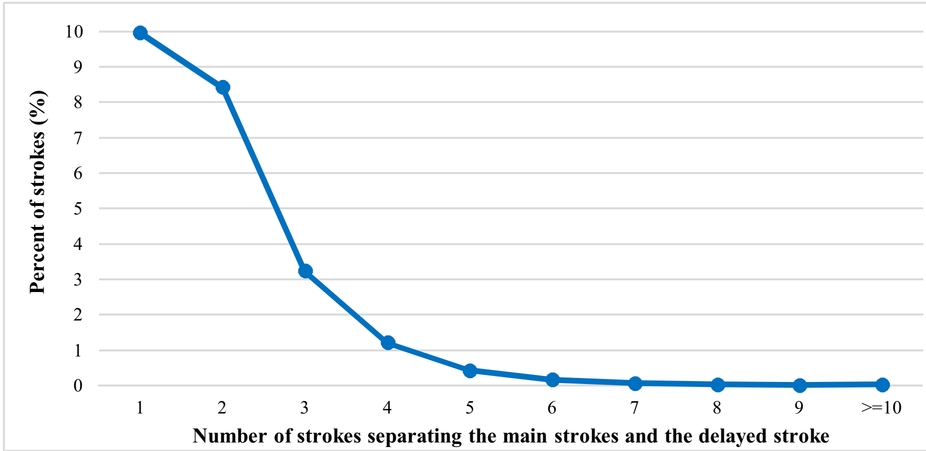


Fig. 13. Distribution of the number of strokes separating the main stroke and the delayed stroke.

(a) ordered: th anh ` con ^ g (b) ordered: t h anh ` cong ^

Fig. 14. Delayed strokes written in different stroke orders.

the trained networks giving us a strong model for labeling the sequence.

For training RNNs, however, the training data need to be pre-segmented and the labeling results need to be post-processed. Therefore, many previous studies combined HMMs and RNNs together to overcome the weak points of each approach. According to [36], CTC (Connectionist Temporal Classification) helps us to represent the output of RNNs as a probability distribution of all possible label sequences. To transform the network output into probabilities, CTC contains a softmax layer and a 'blank' label denoting no label. The CTC layer is usually trained by the gradient descent algorithm.

In order to obtain an input sequence, we extract the features at each pen point of an online handwritten text pattern from VNOnDB. In our experiments, the input for the network is an M x N matrix, with M being the length of the input sequence, and N being the length of the feature vector at each point of the input sequence. Each BLSTM hidden layer consists of two LSTM layers in the opposite directions along the temporal sequence. This architecture has been shown to be successful for online handwritten text recognition in English [37]. The output of the last BLSTM layer is a P x Q matrix, where P is the length of an output sequence and Q is the number of characters in the character set plus one for "blank" ($216+1=217$ classes). Although there are no patterns for 57 categories, we still keep them in the character set so that we can fine-tune the trained networks without retraining from the beginning when we collect patterns for the missing 57 characters.

The output of the last BLSTM layer is fed into a transcription layer for decoding by the best path search algorithm. The final output of the network is a sequence of labels, where each label is a character. In order to evaluate the difference between the output of the network and the ground-truth text, we compute the edit distance (Levenshtein distance), which reflects all insertion, deletion and substitution errors between them. We divide this edit distance by the number of characters in the ground-truth text, and refer to it as the label error rate.

7.2. Short-distance delayed strokes problem

Since DMs are distinctive among vowels in Vietnamese, we need to be able to process delayed strokes caused by DMs. In previous research, some approaches have been tried to deal with these delayed strokes. Liwicki et al. removed all delayed strokes for recognizing English handwritten text patterns, resulting in an improved performance [22]. Another method by Abdelaziz et al. [38] employed character segmentation and rearrangement of delayed strokes for recognizing online Arabic handwritten text. This method depends on the performance of character segmentation and delayed strokes detection.

For Vietnamese, we employ the LSTM network [36], one of the derivative architectures of RNNs, which avoids the vanishing or exploding gradient problem. These problems limit the ordinary RNNs to learn long time-dependent input sequences. Moreover, LSTM seems promising to solve the problem of short-distance delayed strokes as each cell has an internal memory, which facilitates modeling dependencies through time.

As unidirectional networks use the time-series information only from the past or from the future, they have potential limitation for delayed strokes. Therefore, we employ bidirectional LSTM (BLSTM) [37], which is a combination of two opposite-directional LSTM layers. In this way, the BLSTM network is able to achieve long-range context retrieval in both directions of input that is appropriate for online handwritten Vietnamese recognition.

Table 5 shows an example including short-distance delayed strokes, which is recognized by four different recognizers. All of these recognizers are networks with a single layer of 75 RNN or LSTM cells trained by VNOnDB-Line. The bidirectional RNNs and BLSTM networks consist of two directionally opposite layers, each of which has 75 RNN or LSTM cells. The red strokes in the handwritten text pattern are short-distance delayed strokes. The characters in red are results of misrecognized labels. Recognizers based on unidirectional and bidirectional RNN do not recognize them because they have the problem of vanishing and exploding gradient. The unidirectional LSTM recognizes consonants well but it does not recognize vowels with DMs. Only BLSTM recognizes all the characters.

Table 6 shows another experiment on long-distance delayed strokes for the BLSTM recognizer in which the DMs are written after a line is finished. BLSTM recognizes correctly consonants and some vowels without DMs but it does not recognize vowels with DMs written as long-distance delayed strokes. The reason for this phenomenon is that the distances between the main vowel strokes and DM strokes are longer for BLSTM to recognize them correctly.

Table 5
Experiment on Short-distance delayed strokes using different neural networks.

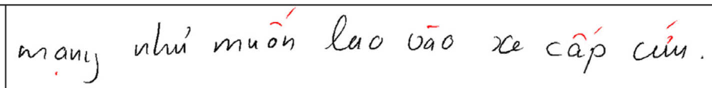
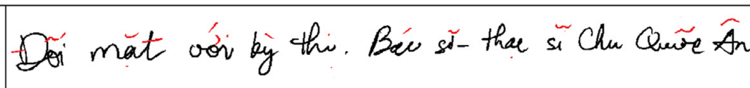
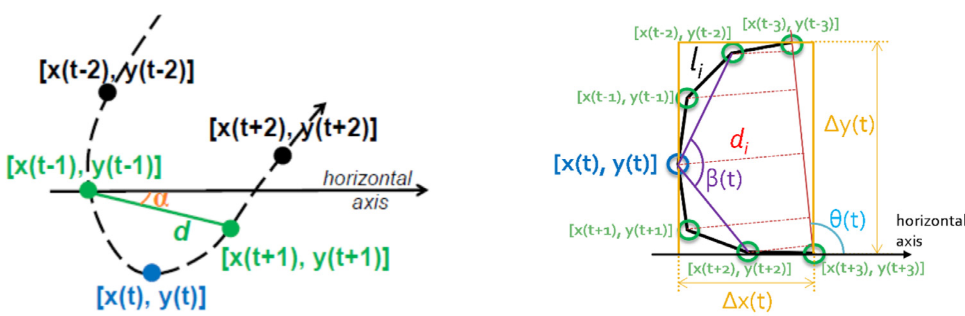
Handwritten text pattern	
Ground-truth text	m ạ n g sp n h ư sp m u ố n sp l a o sp v à o sp x e sp c ấ p sp c ứ u .
Unidirectional RNN	t g sp sp sp sp sp sp sp đ sp
Bidirectional RNN	n a n sp n h ư sp m ư ô n sp k c sp c a sp sp x a sp c á sp c sp
Unidirectional LSTM	m à n g sp n h ư sp m ư ớ n sp l à o sp v à o sp x e sp c á i p sp c ũ u
BLSTM	m ạ n g sp n h ư sp m u ố n sp l a o sp v à o sp x e sp c ấ p sp c ứ u .

Table 6
Experiment on Long-distance delayed strokes using BLSTM.

Handwritten text pattern	
Ground-truth text	Đối mặt với kỳ thị. Bác sĩ - thạc sĩ Chu Quốc Ân,
BLSTM	ới mah coi ly tho. Bao si - thac ái Chu Quoe đa, c - X v . n



(a) F1 features (b) F2 features

Fig. 15. Illustration of the online features (F1 and F2 features).

For this BLSTM, the label error rate on the test set is 10.59%, with 37% of this error rate being caused by the characters with DMs. According to the results of Tables 5 and 6, BLSTM solves the problem of short-distance delayed strokes. Therefore, we use it for all the experiments hereafter.

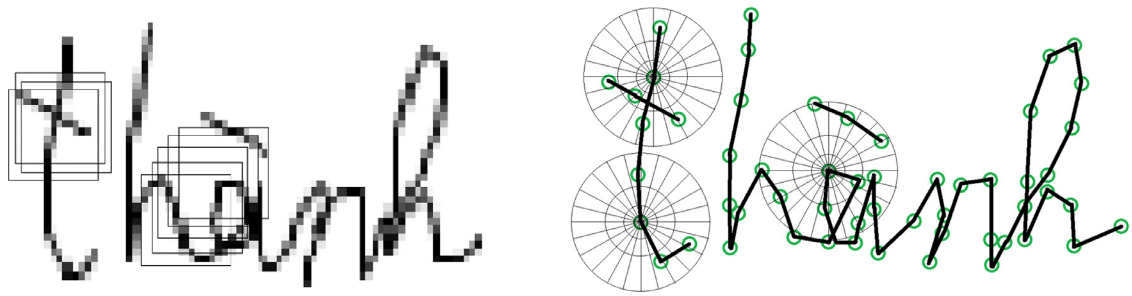
7.3. Features extraction and normalization

Due to the sampling rate of 120 Hz and the resolution of 0.01 mm, there are many redundant points. After some experiments, we apply re-sampling process, or decimation with a factor of two, which eliminates half the points for every stroke. In this re-sampling process, we do not scale to a fixed space gap between consecutive points for getting rid of speed variations, as our purpose is to reduce the amount of computation.

For online handwriting recognition, various features have been studied [22,39]. Some commonly employed features are spatial features, such as distance and x- and y- differences between two adjacent coordinates, pen up/down, curvature at each point and so on. Aspect and curliness of trajectory, stroke slope and linearity have been applied as well. All of these features are point-based features, i.e., they are extracted from each point of pen trajectory; they are also known as local features.

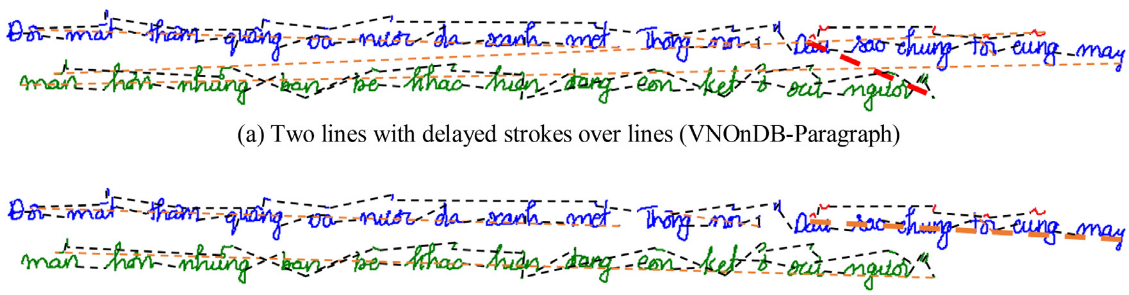
In our first series of experiments in [24], however, we employed only the most basic four dimensional features (F1) from each point: distance between the preceding and succeeding points of the current point (d), writing direction ($\sin(\alpha)$ and $\cos(\alpha)$), and pen-up/pen-down information as illustrated in Fig. 15(a). Among these four features, the feature (d) is useful to discriminate the similar lower case and upper case characters: for instance, ‘c’ versus ‘C’ and ‘o’ versus ‘O’ . Two reasons for using only these four features are: 1) to test the usefulness of VNOnDB, and 2) to find out the extent to which online handwritten patterns in Vietnamese can be recognized by simple features.

Then, the set of features are enlarged in the second series of experiments. The online features (F2) are added for each point related to its vicinity of V points with $V=7$ for the experiments. Fig. 15(b) shows the vicinity of the pen point at time t: it is from the pen point t-3 to the pen point $t+3$ denoted as point $i=1\ldots V$. The first feature of F2 is curvature ($\sin(\beta(t))$ and $\cos(\beta(t))$) where $\beta(t)$ is the angle formed by the sequence of three pen points at times t-2, t, and $t+2$. The second feature relates to the aspect $A(t)=(\Delta y(t)-\Delta x(t))/(\Delta y(t)+\Delta x(t))$, where $\Delta x(t)$, $\Delta y(t)$ are the width and the height of the bounding box of this vicinity. The third feature is curliness $C(t)=L(t)/\max(\Delta x(t),\Delta y(t))$, where l_j is



(a) F3 features (b) F4 features

Fig. 16. Illustration of the offline features (F3 and F4 features).



(a) Two lines with delayed strokes over lines (VNOnDB-Paragraph)

(b) Reordered delayed strokes over lines (VNOnDB-Line)

Fig. 17. The difference of delayed strokes over line between VNOnDB-Paragraph and VNOnDB-Line.

the length between two adjacent points j and $j+1$ with $j=1...V$-1 and $\mathrm{L}(t)=\mathrm{sum}(l_j)$. The fourth feature is linearity $\mathrm{LN}(t)=\mathrm{sum}(d_i{}^2)/V$, where d_i is the distance from point i to the straight line connecting the first and the last point of this vicinity. The last feature is slope ($\sin(\theta(t))$ and $\cos(\theta(t))$), where $\theta(t)$ is the angle formed by the horizontal axis and the straight line connecting the first and the last point of this vicinity. Thus, the F2 feature vector has 7 dimensions, namely $\sin(\beta(t))$, $\cos(\beta(t))$, $\mathrm{A}(t)$, $\mathrm{C}(t)$, $\mathrm{LN}(t)$, $\sin(\theta(t))$ and $\cos(\theta(t))$.

Moreover, two established features, namely offline context bitmap (F3) and online shape context (F4), are extracted. For the F3 features, we compute the number of points belonging to each cell of a square mask (k x k cells, with each cell having b x b bitmaps) [22]. In our experiments, we set k to be 15 and b to be 3. Then, a 5×5 Gaussian filter is employed on the extracted F3 features to absorb positional variation. Consequently, the 15×15 context bitmap is scaled down into a 5×5 matrix by the non-overlap 3×3 average filter in order to reduce the feature dimensions. Thus, the F3 feature vector has 25 dimensions. Fig. 16(a) illustrates the extraction process for F3 features. For F4 features, we count the number of points belonging to each bin of a fan-shaped mask [40], which is illustrated in Fig. 16(b). In our experiments, we set the radius of the largest circle as $r=\mathrm{e}^p$ HIMETRIC units with $p=7$, and divide it into L concentric circles of the radii $r_l=\mathrm{e}^{(l*p)/L}$ with $l=1...L$ ($L=6$), and B radial lines of equal angular distance with $B=24$ with the result of 6×24 bins. Then, a 5-concentric and 5-radial Gaussian filter is applied on the extracted F4 features. Finally, the non-overlap 2-concentric and 3-radial average filter is employed on the 6-concentric and 24-radial masks to obtain 3-concentric and 8-radial masks, which reduces the feature dimensions. Hence, the F4 feature vector has 24 dimensions.

These sizes, resolutions and partitions of the masks for the context bitmap and the shape context are defined by preliminary experiments on the training set. Theoretically, these parameters should only be defined for size-normalized patterns. In our experiments, however, we employ fixed values without size normalization, as almost all the handwritten characters have the size of 40×40 to 50×50 pixels in bitmap and from 800×800 to 1000×1000 in HIMETRIC units.

The extracted features are standardized to zero-mean and unit-variance before the training phase in order to increase the convergence rate by the gradient descent method and keep the balance of contribution of different features. If one of the features has a large range of values, it ends up controlling the distance. We also combine some features together by concatenating them at every point, for example, the F1 features and the F2 features denoted by F1 + F2.

7.4. Experiments and results

In our experiments, the database was divided into three separate sets for training, validation, and testing without writer dependence. The training set contained handwritten texts from 105 writers (153 files), the validation set contained handwritten texts from 35 writers (38 files) and the testing set contained handwritten texts from 60 writers (64 files). There were 289,340 strokes in the training set, 87,306 strokes in the validation set and 104,092 strokes in the testing set.

Then, we conducted experiments on both versions of VNOnDB. We applied BLSTM for each paragraph in VNOnDB-Paragraph, and we applied it for each line in VNOnDB-Line. The training time was reduced for VNOnDB-Line as each text line is much shorter than a paragraph. Moreover, we treated delayed strokes over text lines in VNOnDB-Paragraph as they were, but in VNOnDB-Line as those from the end of each line. Fig. 17 shows an example.

Fig. 17(a) illustrates two lines with delayed strokes over a line in which the blue strokes belong to the first line, the green strokes belong to the second line, and the red strokes are delayed strokes over the line. The black straight dashed lines are off-strokes from left to right, the orange dashed lines are off-strokes from right to left, and the thick red dashed line is an off-stroke to the previous line. In this example, the diacritical marks are written after the sentence is finished. For the first sentence, after writing the period mark in the middle of the first line, the pen is moved back

Table 7
Results of handwritten text recognition at Paragraph level using F1 features (VNOnDB-Paragraph).

Network Architecture	Best Label Error Rate (%)		Training time/epoch (minutes)
	Training set	Testing set	
1 hidden layer			
50-BLSTM	9.96	12.03	55.91
75-BLSTM	11.85	15.22	68.65
100-BLSTM	7.78	12.41	82.47
150-BLSTM	11.73	17.67	120.47
2 hidden layers			
16–32-BLSTM	12.29	12.19	48.71
32–64-BLSTM	6.14	8.45	70.81
64–128-BLSTM	4.07	**7.68**	138.96

Table 8
Results of handwritten text recognition at Line level using F1 features (VNOnDB-Line).

Network Architecture	Best Label Error Rate (%)		Training time/epoch (minutes)
	Training set	Testing set	
1 hidden layer			
50-BLSTM	7.46	11.46	22.01
75-BLSTM	7.68	10.59	38.09
100-BLSTM	6.99	11.57	56.77
150-BLSTM	4.73	10.48	81.60
2 hidden layers			
16–32-BLSTM	10.00	10.59	15.07
32–64-BLSTM	6.31	8.27	28.75
64–128-BLSTM	3.82	**7.51**	66.90

to write the diacritical mark '^' of the first word and so on. Then, the second sentence is written in two lines. After writing the last period mark, the pen is moved back to the previous line to write the diacritical marks '^' and '~' of the word 'Dẫu', and the diacritical marks of the other words 'chúng', 'tôi', 'cũng'. Next, the pen is moved to the second line to write the diacritical marks of the remaining words of the sentence. The pen trajectories are recorded as they are in VNOnDB-Paragraph.

Since the delayed strokes over the line belong to the previous line, the red strokes (the delayed strokes over the line) are reordered to the end of the first line when we segment these patterns into two lines for VNOnDB-Line. Consequently, the thick red straight line in Fig. 17(a) is replaced by another thick orange straight line as shown in Fig. 17(b), which shows that at the end of the first line, the diacritical marks '^' and '~' of the word 'Dẫu', and diacritical marks of other words 'chúng', 'tôi', 'cũng' are written after the last word 'may'. For the second line, the diacritical marks of the word 'mắn' are written after the last period mark, and so on.

After employing the extracted features, several different configurations of networks using one hidden layer and two hidden layers were studied. Each kind of network was trained and tested using a different number of memory cells of the BLSTM hidden layer. In all our experiments, we used the learning rate of 1e-4, the momentum of 0.9 and the early stop scheme after 20 epochs without improving accuracy on the validation set. In these experiments, we applied neither language models nor grammar rules in the post-processing step. For each configuration of the network, we trained it four times with different initial weights and chose the best network for the validation set. The performance on the testing set is shown in Tables 7–10. In these tables, *NC*-BLSTM denotes BLSTM formed by two LSTM layers of *NC* cells. Similarly, *NC1*-*NC2*-BLSTM denotes two-layered BLSTM with the first BLSTM formed by LSTM layers of *NC1* cells and the second BLSTM formed by LSTM layers of *NC2* cells.

Table 7 shows that the longest training time is 138.96 min. and the shortest one is 48.71 min. per epoch on the Intel® X® CPU E5-2630 v2 2.6 GHz with 32GB memory. The best label error rate achieved by the networks is 7.68% when using 2-hidden layer BLSTM network. In general, multi-layer networks perform better than single-layer networks with the same number of parameters. Fig. 18 shows how the label error rates decrease by different networks during the training phase on VNOnDB-Paragraph. Although the 2-hidden layer BLSTM networks converge slower than the 1-hidden layer BLSTM networks, they achieve lower label error rates.

Since VNOnDB-Line consists of handwritten text patterns at the line level, these patterns are shorter than the patterns of VNOnDB-Paragraph, which incurs less processing cost for the recognizer. Table 8 shows that the longest training time per epoch is 66.90 min. and the shortest one is 22.01 min. for VNOnDB-Line, which is about two times faster than for VNOnDB-Paragraph. Moreover, each sequence in VNOnDB-Line contains no delayed strokes over lines, which is easier for training the recognizer. Therefore, the performance of BLSTM on VNOnDB-Line is generally better than on VNOnDB-Paragraph with the same configuration.

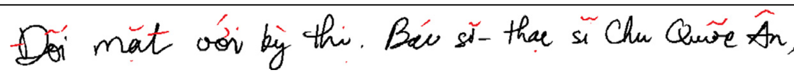
Table 9 shows the label error rate (%) of different networks with different feature sets on the VNOnDB-Line dataset. In the case of one hidden layer networks, the 150-BLSTM network with all features (F1 + F2 + F3 + F4) achieved the best label error rate of 9.92%. In the case of two hidden layer networks, the 64–128 BLSTM with all features achieved the best label error rate of 7.17%.

Table 10 presents recognition results by the 75-BLSTM networks with different feature sets on a sample, which includes many long-distance delayed strokes. In this table, the F4 features have the label error rate (49.02%) better than the other feature sets (49.02% vs around 70%), which shows that the offline features are stable with respect to delayed strokes. In this paper, we focus on delayed strokes since they cause a serious problem for online handwriting recognition for Vietnamese and other languages. In misrecognized samples shown in Table 10, however, errors due to other reasons also appear. These misrecognized samples are usually related to cursively written characters. For example, the third word of the handwritten text pattern in Table 10 without the diacritical marks is 'voi' that was written cursively. The BLSTM networks output the

Table 9
Label error rate (%) with different Sets of Features on VNOnDB-Line.

Network Architecture\Features (Dimensions)	F1 (4)	F2 (7)	F3 (25)	F4 (24)	F1 + F2 (11)	F1 + F3 (29)	F1 + F4 (28)	F1 + F2 + F3 + F4 (60)
1 hidden layer								
50-BLSTM	11.46	16.99	22.88	20.42	12.55	12.69	11.23	13.28
75-BLSTM	10.59	17.97	19.36	15.96	11.44	11.10	11.99	10.14
100-BLSTM	11.57	16.80	19.33	15.55	11.35	10.32	10.15	10.88
150-BLSTM	10.48	16.01	20.27	14.49	11.20	11.71	11.02	**9.92**
2 hidden layers								
16–32-BLSTM	10.59	20.23	29.10	20.88	13.54	13.72	10.31	11.43
32–64-BLSTM	8.27	12.66	17.84	13.89	8.95	9.69	9.42	8.45
64–128-BLSTM	7.51	12.44	14.52	11.32	7.75	7.75	7.72	**7.17**

Table 10
Experiments on Long-distance delayed strokes using different Sets of Features.

Handwritten text pattern		Label Error Rate (%)
Ground-truth text	Đối mặt với kỳ thị . Bác sĩ - thạc sĩ Chu Quốc Ân,	
75-BLSTM (F1)	ới mah coi ly tho. Bao si - thac ái Chu Quoe đa, c - X v . n	72.55
75-BLSTM (F2)	TĐé ma 0ơi Đ thoe Be csi - thac si Ch Cuuo .c1t.... .	74.51
75-BLSTM (F3)	t uT Tờ tì hờ h l iic v .: v	68.63
75-BLSTM (F4)	Đới miậc cn ý th. Beo s thạc 3i ch cuốt Ấn	**49.02**
75-BLSTM (F1 + F2)	Đbi mn can tg thu, Ba si- tha si ch Qoo Đư vV .I. . v nm	78.43
75-BLSTM (F1 + F3)	pi rrah 0ho t th. Bu. si- Tho si Ch oc đ, " "T.: . "n	76.47
75-BLSTM (F1 + F4)	i ml va k th. Ba si thc s Ch Qua aế ặ ớẫm ẫầ	54.90
75-BLSTM (F1 + F2 + F3 + F4)	Đi mnah vo tý th. ae siu thạcái Che Qư đa	50.98

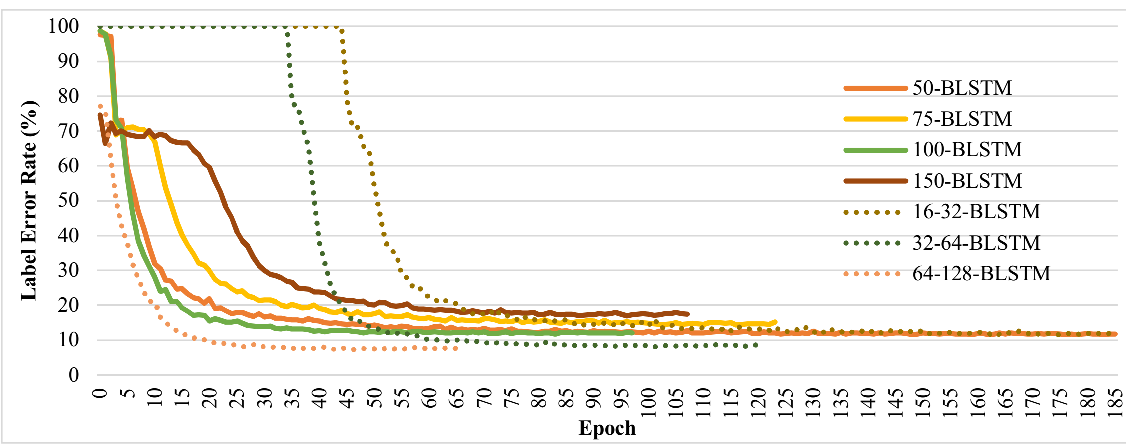


Fig. 18. The label error rate of different BLSTM networks on the testing set of VNOnDB-Paragraph during the training process.

recognition results of this word as ‘coi’, ‘can’, ‘va’ or ‘vo’. Another example is the fifth word in Table X, which is ‘thi’ without the diacritical marks. This word is also written cursively, and the BLSTM networks misrecognize it as ‘tho’, ‘th’ or ‘thu’.

7.5. Evaluations and discussions (suggestions for design of recognizers)

Although 24% of all the strokes contained in VNOnDB are delayed strokes, with 18.5% being short-distance delayed strokes and 5.2% being long-distance delayed strokes, Tables 7 and 8 show that more than 90% are recognized. These results suggest that BLSTM is able to process delayed strokes. Considering the results shown in Tables 5 and 6 together, BLSTM seems to process short-distance delayed strokes but still has difficulty in recognizing long-distance delayed strokes. According to Table 9, the F1 features are useful for handwriting recognition although they are quite simple. One approach to solve this problem is by detecting delayed strokes, relocating them near the proper characters and applying BLSTM. An alternative method could be to employ offline recognition to a text-line image produced from a text line of strokes by discarding temporal information. Trying out these methods and comparing them with the method proposed here is left for future research.

8. Conclusion

This paper presented our research on creating a Vietnamese Online Handwriting Database and analyzing the collected patterns. We have published VNOnDB to TC-11 website at http://tc11.cvc.uab.es/datasets/HANDS-VNOnDB_1/ so that this would be useful for all the people working on handwriting recognition. We studied the Vietnamese online handwriting recognition using the state-of-the-art recognition methods with different features. Although nearly 24% of the strokes in the database seem to be delayed strokes, the best recognizer by BLSTM achieved an accuracy of over 90%. This shows that BLSTM is capable of processing delayed strokes to some extent. However, this performance needs to be improved further for real-world applications by collecting more patterns, employing language models, combining online and offline recognition methods and so on. Besides collecting more patterns, we still need to add character-level ground truth in order to analyze more thoroughly how short-distance and long-distance delayed strokes affect the recognizers. Moreover, we are studying additional features and modifying the recognizers to solve the delayed-stroke problem. In future, we plan to introduce a language model with the recognition methods and combine online recognition methods and offline recognition methods as we did for online handwritten Japanese recognition [41].

Acknowledgment

We would like to express our thanks to all individuals who have contributed to the Vietnamese Online Handwriting Database. Thanks are also due to Professor Dang Duc Trong, Dean of Department of Mathematics and Computer Sciences, Ho Chi Minh City University of Science, Vietnam who helped us collect most of the patterns in VNOnDB from the students. Thanks are also due to Prof. B Indurkhya for discussing the recognition method and improving the presentation.

References

[1] I. Guyon, L. Schomaker, UNIPEN project of on-line data exchange and recognizer benchmarks, in: Proceedings of the 12th IAPR International Conference Pattern Recognition, Jerusalem, 1994, pp. 29–33. Vol. 3 - Conf. C Signal Process. (Cat. No.94CH3440-5). doi: 10.1109/ICPR.1994.576870.
[2] C. Viard-Gaudin, P.M. Lallican, S. Knerr, P. Binter, The IRESTE On/Off (IRONOFF) dual handwriting database, in: Proceedings of the Fifth International Conference on Document Analysis and Recognition ICDAR '99, 1999, pp. 455–458, doi:10.1109/ICDAR.1999.791823.
[3] M. Liwicki, H. Bunke, IAM-OnDB - An on-line English sentence database acquired from handwritten text on a whiteboard, in: Proceedings of the International Conference on Document Analysis and Recognition ICDAR, IEEE, 2005, pp. 956–961, doi:10.1109/ICDAR.2005.132.
[4] A. Shivram, C. Ramaiah, S. Setlur, V. Govindaraju, IBM-UB-1: A dual mode unconstrained english handwriting dataset, in: Proceedings of the International Conference on Document Analysis and Recognition, 2013, pp. 13–17, doi:10.1109/ICDAR.2013.12.
[5] M. Nakagawa, K. Matsumoto, Collection of on-line handwritten Japanese character pattern databases and their analyses, Int. J. Doc. Anal. Recognit. 7 (2004) 69–81, doi:10.1007/s10032-004-0125-4.
[6] M. Kherallah, N. Tagougui, A.M. Alimi, H. El Abed, V. Märgner, Online Arabic handwriting recognition competition, in: Proceedings of the International Conference on Document Analysis and Recognition, 2011, pp. 1454–1458, doi:10.1109/ICDAR.2011.289.
[7] C.L. Liu, F. Yin, D.H. Wang, Q.F. Wang, CASIA online and offline Chinese handwriting databases, in: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2011, pp. 37–41, doi:10.1109/ICDAR.2011.17.
[8] L. Jin, Y. Gao, G. Liu, Y. Li, K. Ding, SCUT-COUCH2009—a comprehensive online unconstrained Chinese handwriting database and benchmark evaluation, Int. J. Doc. Anal. Recognit. 14 (2011) 53–64, doi:10.1007/s10032-010-0116-6.
[9] X.-Y. Zhang, Y. Bengio, C.-L. Liu, Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark, Pattern Recognit 61 (2016) 348–360, doi:10.1016/j.patcog.2016.08.005.
[10] H. Mouchère, C. Viard-gaudin, D.H. Kim, J.H. Kim, U. Garai, CROHME2011 : Competition on recognition of online handwritten mathematical expressions, in: Proceedings of the International Conference on Document Analysis and Recognition, 2011, pp. 1497–1500, doi:10.1109/ICDAR.2011.297.
[11] R. Plamondon, S.N. Srihari, Online and off-line handwriting recognition: a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 63–84, doi:10.1109/34.824821.
[12] C.L. Liu, S. Jaeger, M. Nakagawa, Online recognition of chinese characters: the state-of-the-art, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 198–213, doi:10.1109/TPAMI.2004.1262182.
[13] J. Sternby, J. Morwing, J. Andersson, C. Friberg, On-line Arabic handwriting recognition with template, Pattern Recognit 42 (2009) 3278–3286. https://doi.org/10.1016/j.patcog.2008.12.017.
[14] T. Mondal, U. Bhattacharya, S.K. Parui, K. Das, D. Mandalapu, On-line handwriting recognition of Indian scripts - The first benchmark, in: Proceedings of the 12th International Conference on Frontiers Handwriting Recognition, ICFHR 2010, 2010, pp. 200–205, doi:10.1109/ICFHR.2010.39.
[15] H.A. Al Hamad, R. Abu Zitar, Development of an efficient neural-based segmentation technique for Arabic handwriting recognition, Pattern Recognit. 43 (2010) 2773–2798, doi:10.1016/j.patcog.2010.03.005.
[16] B. Zhu, M. Nakagawa, in: X. Ding (Ed.), Online Handwritten Chinese/Japanese Character Recognition, Adv. Character Recognit., InTech, 2012, pp. 51–68, doi:10.5772/51474.
[17] B. Zhu, M. Nakagawa, A robust method for coarse classifier construction from a large number of basic recognizers for on-line handwritten Chinese/Japanese character recognition, Pattern Recognit 47 (2014) 685–693. https://doi.org/10.1016/j.patcog.2013.08.011.
[18] F. Álvaro, J.A. Sánchez, J.M. Benedí, An integrated grammar-based approach for mathematical expression recognition, Pattern Recognit. 51 (2016) 135–147, doi:10.1016/j.patcog.2015.09.013.
[19] P.A. Phuong, N.Q. Tao, L.C. Mai, An efficient model for isolated Vietnamese handwritten recognition, in: Proceedings of the 4th International Conference on I Intelligent Information Hiding and Multimedia Signal Processing. IIH-MSP 2008, 2008, pp. 358–361, doi:10.1109/IIH-MSP.2008.67.
[20] D.K. Nguyen, T.D. Bui, On the problem of classifying vietnamese online handwritten characters, in: Proceedings of the 10th International Conference on Control Automation Robotics & Vision, 2008, pp. 803–808, doi:10.1109/ICARCV.2008.4795620.
[21] DeCao Tran, An efficient method for on-line Vietnamese handwritten character recognition, in: Proceedings of the Third International Symposium on Information and Communication Technology, 2012, pp. 135–141, doi:10.1145/2350716.2350737.
[22] S. Jaeger, S. Manke, J. Reichert, A. Waibel, Online handwriting recognition: The NPen++ recognizer, Int. J. Doc. Anal. Recognit. 3 (2001) 169–180, doi:10.1007/PL00013559.
[23] S.J. Smith, M.O. Bourgoin, K. Sims, H.L. Voorhees, Handwritten character classification using nearest neighbor in large databases, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 915–919, doi:10.1109/34.310689.
[24] H.T. Nguyen, C.T. Nguyen, P.T. Bao, M. Nakagawa, Preparation of an unconstrained vietnamese online handwriting database and recognition experiments by recurrent neural networks, in: Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition, IEEE, 2016, pp. 144–149, doi:10.1109/ICFHR.2016.0038.
[25] H. Phe, Vietnamese Dictionary, Vietnam Lexicography Centre, Social Sciences Publishing House, Da Nang, 2002.
[26] T.B. Nguyen, T.M.H. Nguyen, L. Romary, X.L. Vu, Developping tools and building linguistic resources for Vietnamese morpho-syntactic processing, in: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004), 2004, pp. 1231–1234. doi:doi:10.1.1.673.4308.
[27] S.M. Watt, T. Underhill, Ink Markup Language (InkML), (2009). https://www.w3.org/TR/InkML/ (accessed May 20, 2009).
[28] O. Samanta, U. Bhattacharya, S.K. Parui, Smoothing of HMM parameters for efficient recognition of online handwriting, Pattern Recognit 47 (2014) 3614–3629. https://doi.org/10.1016/j.patcog.2014.04.019.
[29] P.P. Roy, A.K. Bhunia, A. Das, P. Dey, U. Pal, HMM-based Indic handwritten word recognition using zone segmentation, Pattern Recognit 60 (2016) 1057–1075. https://doi.org/10.1016/j.patcog.2016.04.012.
[30] T. Plötz, G.A. Fink, Markov models for offline handwriting recognition: A survey, Int. J. Doc. Anal. Recognit. 12 (2009) 269–298, doi:10.1007/s10032-009-0098-4.
[31] B. Zhu, M. Nakagawa, Building a compact online MRF recognizer for large character set by structured dictionary representation and vector quantization technique, Pattern Recognit 47 (2014) 982–993, doi:10.1016/j.patcog.2013.09.031.
[32] J. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning, 2001, pp. 282–289.
[33] X. Zhou, Y. Zhang, F. Tian, H. Wang, C. Liu, Minimum-risk training for semi-Markov conditional random fields with application to handwritten Chinese / Japanese text recognition, Pattern Recognit 47 (2014) 1904–1916, doi:10.1016/j.patcog.2013.12.002.
[34] L. Gupta, M. McAvoy, J. Phegley, Classification of temporal sequences via prediction using the simple recurrent neural network, Pattern Recognit 33 (2000) 1759–1770 doi:https://doi.org/10.1016/S0031-3203(99)00149-1.
[35] M. Liwicki, H. Bunke, Combining diverse on-line and off-line systems for handwritten text line recognition, Pattern Recognit 42 (2009) 3254–3263. https://doi.org/10.1016/j.patcog.2008.10.030.
[36] A. Graves, Supervised Sequence Labelling With Recurrent Neural Networks, Preprint, Springer-Verlag, Berlin Heidelberg, 2012, doi:10.1007/978-3-642-24797-2.
[37] M. Liwicki, A. Graves, H. Bunke, J. Schmidhuber, A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks, in: Proceedings of the 9th International Conference on Document Analysis and Recognition, 2007, pp. 367–371. doi:10.1.1.139.5852.
[38] A.H. Abdelaziz, S. Watanabe, J.R. Hershey, E. Vincent, D. Kolossa, Uncertainty Propagation Through Deep Neural Networks, Interspeech, 2015.
[39] F. Álvaro, J.A. Sánchez, J.M. Benedí, Recognition of on-line handwritten mathematical expressions using 2D stochastic context-free grammars and hidden Markov models, Pattern Recognit. Lett. 35 (2014) 58–67, doi:10.1016/j.patrec.2012.09.023.
[40] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 509–522, doi:10.1109/34.993558.
[41] B. Zhu, X.-D. Zhou, C.-L. Liu, M. Nakagawa, A robust model for on-line handwritten japanese text recognition, Int. J. Doc. Anal. Recognit. 13 (2010) 121–131, doi:10.1007/s10032-009-0111-y.

Hung Tuan Nguyen was born on 18 May 1991 in Vietnam. He received his B.Sc. degree in Computer Science from Ho Chi Minh City University of Science in 2013. Since April 2015 he is a M.Sc. student in Department of Computer and Information Sciences at Tokyo University of Agriculture and Technology. He is currently working on handwritten text recognition of Vietnamese and English text.

Cuong Tuan Nguyen was born on 21 March 1988 in Vietnam. He received his B.Eng. degree in Computer Engineering and M.Sc. degree in Computer Science from Ho Chi Minh city University of Technology and Tokyo University of Agriculture and Technology (TUAT), respectively. Since April 2014 he is a Ph.D. student in Department of Electronic and Information Engineering at TUAT. He is currently working on handwritten text recognition of Japanese and English text.

Pham The Bao received the B.Sc. degree from Ho Chi Minh City University, in 1995 and the Ph.D. degree from University of Science, in 2009. He is currently a Professor of Mathematics & Computer Science faculty, University of Science. His current researches focus on image processing, pattern recognition and computing (intelligent computing, biological computing, high-performance computing).

Masaki Nakagawa was born on 31 October 1954 in Japan. He received his B.Sc. and M.Sc. degrees from the University of Tokyo in 1977 and 1979, respectively. During the academic year 1977/78, he enrolled in the Computer Science course at Essex University in England and received the M.Sc. with distinction in Computer Studies in July 1979. He received his Ph.D. in Information Science from University of Tokyo in December 1988. He has been working at Tokyo University of Agriculture and Technology since April 1979. He is currently a Professor of Media Interaction in the Department of Computer and Information Sciences.