Technical Correspondence

End-to-End Online Writer Identification With Recurrent Neural Network

Xu-Yao Zhang, Guo-Sen Xie, Cheng-Lin Liu, and Yoshua Bengio

***Abstract*—Writer identification is an important topic for pattern recognition and artificial intelligence. Traditional methods rely heavily on sophisticated hand-crafted features to represent the characteristics of different writers. In this paper, we propose an end-to-end framework for online text-independent writer identification by using a recurrent neural network (RNN). Specifically, the handwriting data of a particular writer are represented by a set of random hybrid strokes (RHSs). Each RHS is a randomly sampled short sequence representing pen tip movements (xy-coordinates) and pen-down or pen-up states. RHS is independent of the content and language involved in handwriting; therefore, writer identification at the RHS level is more general and convenient than the character level or the word level, which also requires character/word segmentation. The RNN model with bidirectional long short-term memory is used to encode each RHS into a fixed-length vector for final classification. All the RHSs of a writer are classified independently, and then, the posterior probabilities are averaged to make the final decision. The proposed framework is end-to-end and does not require any domain knowledge for handwriting data analysis. Experiments on both English (133 writers) and Chinese (186 writers) databases verify the advantages of our method compared with other state-of-the-art approaches.**

***Index Terms*—End-to-end, long short-term memory (LSTM), online handwriting, recurrent neural network (RNN), writer identification.**

I. Introduction

Nowadays, pen- and touch-based human–machine interactions are becoming more and more popular in our daily life, due to the rapid development of pattern recognition and machine learning technologies. Handwriting is a distinctive and measurable characteristic to label and describe individuals [30]. Contrary to the physiological characteristics [17] (such as fingerprint, palmprint, face, iris, and retina), handwriting is a behavioral characteristic. Writer identification based on handwriting data analysis is an efficient and effective strategy for biometrics [33]. The purpose of writer identification is to determine the genuine writer from a list of registered candidates according to the similarity between their handwritings, which is a challenging problem in artificial intelligence.

According to the type of input data, writer identification systems are divided into online and offline. The offline system is to identify the writer from the scanned image of handwritings, which is more general but difficult to obtain high accuracy. Contrarily, online writer identification exploits much richer (temporal and spatial) information [31] recorded in the writing process such as speed, angle, pressure, and so on. Hence, online writer identification is usually much more accurate. With the rapid development of touchscreen-based or pen-enabled mobile devices [29], online writer identification is becoming more and more popular for personal authentication, digital forensics, and other applications. Moreover, online writer identification also has a close relationship with online signature verification [16], [19], [25].

Writer identification can be implemented either in the text-dependent or text-independent manner [5]. Higher accuracies can be achieved for text-dependent writer identification; however, the practical application of this approach is constrained and limited, due to the requirement of writing with fixed text contents. Text-independent writer identification is much more general and efficient in real applications. However, to achieve high accuracy, this kind of system needs to capture the subtle difference between handwriting styles of different writers and eliminate the influence from the large difference in handwriting contents, which makes text-independent writer identification a challenging problem.

Traditional approaches for writer identification are mostly based on sophisticated hand-crafted features. For example, the textural (contour orientation and curvature) and allographic (character shape) features were designed by Bulacu and Schomaker [5] for writer identification. The interval-valued symbolic features were used by Guru and Prakash [14] to represent the online signatures. In [29], 16 types of histogram-based features were extracted for signature verification. For writer identification with online whiteboard data, more than 45 types of features were designed by Schlapbach *et al.* [31], such as xy-coordinates, direction, acceleration, curvature, stroke length, angle, and so on. The shape primitive representation and hierarchical structure were used by Li *et al.* [21] for online text-independent writer identification. Recently, the convolutional neural network (CNN) has been successfully used for writer identification [37]. However, because CNN is more suitable to deal with fixed-size images, the stroke and character segmentation were required as a preprocessing step, as shown in [38], and hand-crafted features were still needed to achieve high accuracy for CNN-based writer identification systems.

To develop a fully end-to-end writer identification system, we turn to the recurrent neural network (RNN) due to its powerful ability in dealing with sequential data with arbitrary length. This paper is focused on the challenging and important online text-independent writer identification task. First, we represent the online handwriting data of a particular writer by a set of random hybrid strokes (RHSs). Each RHS is a randomly sampled short sequence from the whole online data with

Manuscript received November 14, 2015; revised March 8, 2016 and July 13, 2016; accepted October 22, 2016. Date of publication December 28, 2016; date of current version March 13, 2017. This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDB02060009, and in part by the National Natural Science Foundation of China under Grant 61403380. This paper was recommended by Guest Editor R. Plamondon.

X.-Y. Zhang, and G.-S. Xie are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: xyz@nlpr.ia.ac.cn; guosen.xie@nlpr.ia.ac.cn)

C.-L. Liu is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, and with the CAS Center for Excellence in Brain Science and Intelligence Technology, and also with the University of the Chinese Academy of Sciences, Beijing 100190, China (e-mail: liucl@nlpr.ia.ac.cn)

Y. Bengio is with the Montreal Institute for Learning Algorithms, University of Montreal, Montreal, QC H3T 1J4, Canada (e-mail: yoshua.bengio@umontreal.ca)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2016.2634921

TIME	BUTTONS	ORIENTATION		X	Y	PRESSURE
8162501	0	1230	560	4830	13918	0
8162511	0	1220	560	4845	13950	0
8162521	0	1220	560	4856	13983	0
8162531	0	1220	560	4865	14015	0
8162546	1	1220	560	4865	14015	466
8162556	1	1220	560	4865	14015	515
8162566	1	1220	560	4865	14015	516
8162576	1	1240	550	4865	14015	581
8162586	1	1240	550	4865	14015	627
8162596	1	1250	560	4865	14015	662
8162606	1	1250	560	4865	14015	700
8162616	1	1250	560	4865	14015	711
8162626	1	1250	560	4855	13969	727
......						
8287331	0	1380	600	17840	5982	0
8287328	0	1380	600	17740	6360	0
8287338	0	1380	600	17740	6360	0
End of data						

Fig. 1. Raw data of online handwriting.

both pen-down (real stroke) and pen-up (imaginary stroke) information. The extraction of RHS is very efficient and does not require any domain-specific knowledge. The RNN model with bidirectional long short-term memory (LSTM) [9], [15] is then used to encode the RHS into a fixed-length vector for final classification. Writer identification at the RHS level is much more efficient than the character level or the word level, which requires character/word segmentation. RHS is also independent of the content and language involved in handwriting. Moreover, many RHSs can be sampled from the original data to build a large enough training set for RNN training. At last, all RHSs sampled from one page (or text line) are classified independently and combined to make the ensemble-based decision for final writer identification.

The RNN has been used to achieve amazing performance in different tasks such as image caption generation [36], machine translation [7], speech recognition [13], handwriting generation [10], handwriting recognition [12], [22], and so on. This paper shows another case that very good performance can be achieved by using RNN for the end-to-end writer identification without any domain-specific knowledge of handwriting. On the biometrics ideal test (BIT) database [1], we have achieved 100% accuracy for English and 99.46% accuracy for Chinese writer identifications, which significantly outperform previous approaches. To the best of our knowledge, this is the first work on using RNN for the writer identification task.

The rest of this paper is organized as follows. Section II describes the proposed RHS representation for online handwriting data. Section III introduces the RNN used in our system. Section IV details the whole writer identification process. Section V shows the experimental results, and Section VI draws the concluding remarks.

II. RHS Representation

Online handwriting data are very common in human–machine interactions and can be captured by different electronic devices such as PDA, smartphone, tablet PC, and so on. During the data acquisition process, different signals representing the handwriting movements can be recorded. For example, Fig. 1 shows the raw data for an online handwriting example from [1]. Each line is a sampled point during the writing process, which is characterized by features listed in the following order: time stamp, button status, azimuth, altitude, x-coordinate, y-coordinate, and pressure.

For a general and robust representation of online handwriting data, we only consider the xy-coordinates and the binary pressure information (pen-down or pen-up). Specifically, each data (for a particular writer) are represented as a sequence by

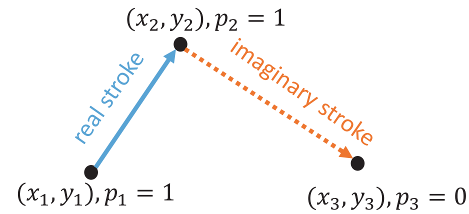

Fig. 2. Illustration of the hybrid strokes.

$$S = [[x_1, y_1, p_1], [x_2, y_2, p_2], \ldots, [x_n, y_n, p_n]] \tag{1}$$

where p_i is the indicator for pen-down ($p_i = 1$) or pen-up ($p_i = 0$). This kind of representation only uses the basic information of online handwriting and, therefore, is applicable for most handwriting-enabled devices. We transform S (point-level) into another sequence (stroke-level), which represents the direction of the pen movement[1]

$$\begin{aligned}\Delta S = [&[x_2 - x_1, y_2 - y_1, p_2 \times p_1], \ldots \\ &[x_i - x_{i-1}, y_i - y_{i-1}, p_i \times p_{i-1}], \ldots \\ &[x_n - x_{n-1}, y_n - y_{n-1}, p_n \times p_{n-1}]].\end{aligned} \tag{2}$$

As shown in Fig. 2, the line connecting two adjacent points can belong to different strokes. When both of the two points are pen-down ($p_i \times p_{i-1} = 1$), it is a real stroke; otherwise, it is an imaginary stroke. The imaginary stroke is recorded during the pen-up period, which can be either within or between different characters or words. Therefore, imaginary stroke is widely used for word/character segmentation. However, we do not care about this, we just use an index of $p_i \times p_{i-1}$ to indicate a real or imaginary stroke, and the RNN model will try to interpret it for the task of writer identification.

Note that ΔS can be a text line or even a text page produced by a particular writer; hence, it should be a very long sequence. In writer identification, each writer may not have sufficient training data (i.e., only one or two ΔS). Therefore, to generate enough training data, we randomly sample multiple short continuous subsequences from ΔS. Each subsequence is denoted as an RHS, because it may include both real and imaginary strokes. In other words, RHS is a short subsequence randomly sampled from ΔS with a general form of

$$\text{RHS} = [\ldots, [\Delta x_i, \Delta y_i, \widetilde{p_i}], \ldots] \tag{3}$$

where $\Delta x_i = x_i - x_{i-1}$, $\Delta y_i = y_i - y_{i-1}$, and $\widetilde{p_i} = p_i \times p_{i-1}$. The RNN classifier is then trained at the RHS level for the task of writer identification. In this paper, we consider end-to-end writer identification. As described above, RHS is extracted from the raw data without any domain knowledge of handwriting. Moreover, RHS is independent of the content and language involved in handwriting and, therefore, can be effectively used for text-independent writer identification.

Note that the term "stroke" has been used in many different contexts in the literature,[2] for example, to describe the segments between pen-down and pen-up, or the segments between points of high curvature, or the segments corresponding to a neuromuscular command. In this paper, the stroke is simply defined at the level of sampling points (difference between two sampling points) and not at the level of longer segments, for which the term "stroke" is typically used.

[1]For some special devices, some preprocessing steps such as a low-pass filter may be helpful for cleaning the signals.

[2]Thank one of the anonymous reviewers for the explanation of this point.

III. Recurrent Neural Network

The RNN is a natural generalization of feedforward neural networks to sequences [35]. Given an input sequence $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k]$ (different samples may have different k), the RNN can be used to encode this variable-length sequence into a fixed-length vector representation. At each time step, a hidden state is produced, resulting in a hidden sequence of $[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_k]$. The activation of the hidden state at time step t is computed as a function f of the current input $\mathbf{x}_t$ and previous hidden state $\mathbf{h}_{t-1}$ as

$$\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{h}_{t-1}). \quad (4)$$

At each time step, an optional output can be produced by $\mathbf{y}_t = g(\mathbf{h}_t)$, resulting in an output sequence $[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k]$, which can be used for sequence-to-sequence tasks (such as speech and handwriting recognition) [11]. In this paper, we consider sequence classification; hence, only one output is produced based on the final hidden state $\mathbf{h}_k$. In other words, the input sequence is encoded into a fixed-length vector $\mathbf{h}_k$, due to the recursively applied transition function f.

A. Long Short-Term Memory

The RNN maintains activations for each time step, which makes RNN to be extremely deep. Therefore, the recurrent unit f is very important to guarantee the success of RNN. LSTM [8], [15] is widely used because it is resistant to the vanishing gradient problem and can learn long-term dependence. In LSTM, for time step t, there is an input gate $\mathbf{i}_t$, forget gate $\mathbf{f}_t$, and output gate $\mathbf{o}_t$

$$\mathbf{i}_t = \mathrm{sigm}\left(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + b_i\right) \quad (5)$$

$$\mathbf{f}_t = \mathrm{sigm}\left(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + b_f\right) \quad (6)$$

$$\mathbf{o}_t = \mathrm{sigm}\left(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + b_o\right) \quad (7)$$

$$\widetilde{\mathbf{c}_t} = \tanh\left(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + b_c\right) \quad (8)$$

$$\mathbf{c}_t = \mathbf{i}_t \odot \widetilde{\mathbf{c}_t} + \mathbf{f}_t \odot \mathbf{c}_{t-1} \quad (9)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (10)$$

where W_* is the input-to-hidden weight matrix, U_* is the state-to-state recurrent weight matrix, and b_* is the bias vector. The operation $\odot$ denotes the elementwise vector product. The hidden state of LSTM is the concatenation of $(\mathbf{c}_t, \mathbf{h}_t)$. The long-term memory is saved in $\mathbf{c}_t$, and the forget gate and input gate are used to control the updating of $\mathbf{c}_t$, as shown in (9), while the output gate is used to control the updating of $\mathbf{h}_t$, as shown in (10).

B. Bidirectional RNN

The RNN is a directional model that only uses past contexts. However, in real applications, contexts from both past and future are useful and complementary to each other [12]. Therefore, we combine forward (left to right) and backward (right to left) LSTM layers to build a bidirectional RNN model [32]. As shown in Fig. 3, by passing $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k]$ through the forward LSTM, we can obtain a hidden state sequence of $[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_k]$. Meanwhile, by passing the reversed sequence of $[\mathbf{x}_k, \mathbf{x}_{k-1}, \ldots, \mathbf{x}_1]$ through the backward LSTM, we can obtain another hidden state sequence of $[\mathbf{h}'_1, \mathbf{h}'_2, \ldots, \mathbf{h}'_k]$. The $\mathbf{h}_k$ and $\mathbf{h}'_k$ can be viewed as summaries of the input sequence in forward and backward directions. Therefore, we combine them to obtain a fixed-length representation for the input sequence

$$\text{Fixed Length Feature} = \mathbf{h}_k + \mathbf{h}'_k \quad (11)$$

which is then fed into a logistic regression layer for final classification (writer identification). The whole model can be efficiently and effectively trained by minimizing the multiclass negative log-likelihood loss with the backpropagation algorithm [28].

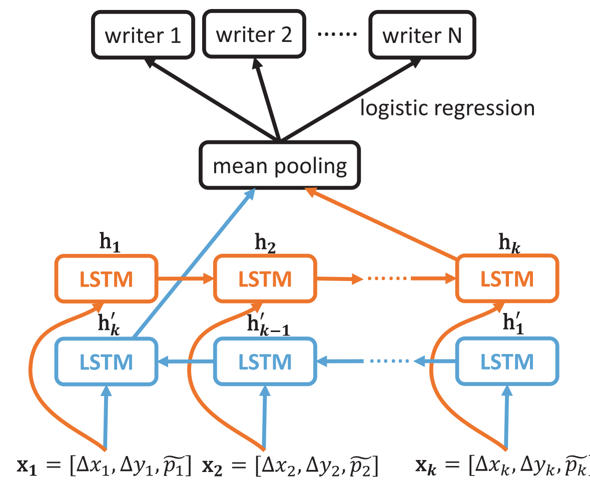

Fig. 3. Bidirectional RNN model for writer identification.

C. Initialization and Optimization

Initialization is very important for deep neural networks. We initialize all the weight matrices in LSTM (W_* and U_*) and logistic regression by random values drawn from the zero-mean Gaussian distribution with standard deviation 0.01. All bias terms are initialized as zeros, except the forget gate in LSTM. As suggested by Jozefowicz *et al.* [18], we initialize the forget gate bias b_f to be a large value of 5. The purpose of doing so is to guarantee the forget gate in (6) being initialized close to one (which means no forgetting), and then, long-range dependencies can be better learned in the beginning of training. The cell and hidden states of LSTMs are initialized as zeros. Optimization is another important issue for deep learning. In this paper, we use a recently proposed first-order gradient method called Adam [20], which is based on adaptive estimation of lower order moments. These strategies make the training of RNN to be both efficient and effective.

IV. Writer Identification With RNN

In writer identification, each writer should first register in the system by showing some handwriting data. Each data are represented by a sequence, as shown in (1), which can be the record of a text line or a text page. We further transform (1) into another sequence, as shown in (2), for better representation, which ignores the absolute positions of pen and captures the pen moving directions. There is an index in (2) to indicate a real or imaginary stroke. A major challenge for writer identification is that each writer may have very small number of training data. For example, each writer may only have one text line or text page. To handle this problem, we randomly sample multiple short subsequences, which is denoted as RHS, as shown in (3).

We can randomly sample as many RHSs as we need to build a large enough training dataset for RNN training. In the testing process, as shown in Fig. 4, multiple RHSs are randomly sampled from the raw data of a particular writer, and then, all RHSs are classified independently by the pretrained RNN model (as described in Section III). At last, the posterior probabilities of all RHSs are added together to give an ensemble-based prediction, which is the final writer identification result. A single RHS is too simple to achieve a high accuracy for writer identification; however, the ensemble of many randomly sampled RHSs can efficiently and significantly boost the performance.

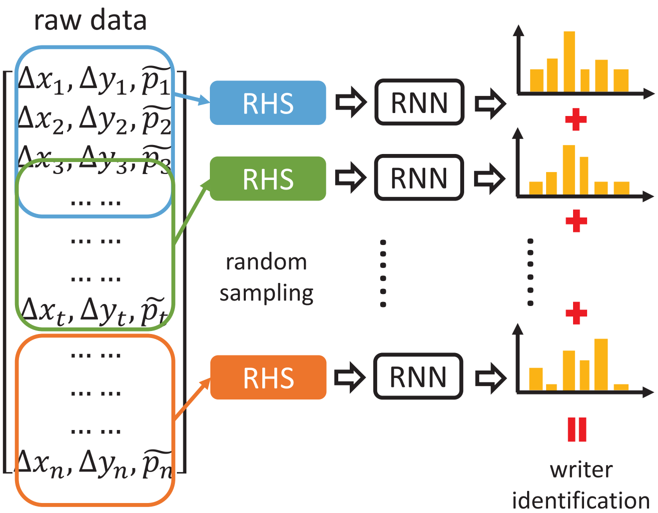

Fig. 4. Illustration of the proposed end-to-end writer identification system. From the raw data, many RHSs are randomly sampled. After that, each RHS is fed into the RNN model individually to produce a posterior probability histogram (for different writers). At last, all the histograms are averaged to make the ensemble-based decision.

Each RHS is not a genuine "stroke" in handwriting, it is only a short sequence containing both real and imaginary strokes (see Section II). Each element in an RHS sequence is a simple 3-D vector. We do not use any domain knowledge of handwriting such as character or word segmentation. The handwriting style of a particular writer is implicitly embedded in RHS, but we do not know what it is. Fortunately, the RNN model can find such characteristic by intelligently deciding whether to memorize or forget a particular stroke in RHS. Moreover, the parameters involved in LSTMs are automatically tuned to extract discriminative features from RHS for the task of writer identification.

As shown in Fig. 4, what we proposed is an end-to-end writer identification system that is operated directly on raw data. Compared with previous hand-crafted feature-based approaches [5], [14], [29], [31], our method is much more general and efficient, which can eliminate the need of complicated human-based preprocessing and feature engineering. Moreover, due to the end-to-end training, sufficient randomly sampled training data, and ensemble-based decision making, our method can achieve significantly better performance compared with previous state-of-the-art approaches.

V. Experiments

In this section, we will compare our method with other state-of-the-art approaches for both English and Chinese writer identifications. Moreover, we also give some analyses on different aspects of our method. At last, we show the performance of cross-language writer identification.

A. Database

In this section, we conduct experiments to compare our approach with other previously proposed writer identification methods. The used dataset is the handwriting database from the BIT [1]. The samples in this database are collected by a Wacom Intuos2 tablet; therefore, rich sequential online handwriting information has been recorded, as shown in Fig. 1. The database is written in the language of both English and Chinese. We compare different methods on two datasets. The first dataset is written in Chinese by 187 writers, while the second dataset is written in English by 134 writers. We use the same experimental setting as [38]: two free-content pages from each writer are used for training and one fixed-content page from the same writer is used for testing. During our experimental process, we find that there are two writers in the database containing exactly the same information. This may be caused by some mistakes in data collection. Therefore, we delete the duplicated writer from the database, resulting in 186 writers for Chinese and 133 writers for English datasets, respectively.

B. Implementation Details

Each handwritten text page for a writer is represented by a long sequence, as shown in (2). The sequence length is variable due to different writing speeds and contents for different writers. The average sequence length (number of sampling points) for each text page of all writers is 6531 for the English dataset and 8978 for the Chinese dataset. For each text page (sequence), we randomly sample 1000 RHSs from it. Different RHSs may have overlap in the original sequence (see Fig. 4). The length of each RHS is fixed as 100. Therefore, for each writer, we have 2000 training samples (two text pages) and 1000 test samples (one text page), which are large enough to obtain a high accuracy.

For the bidirectional RNN model shown in Fig. 3, the dimensionality for the hidden state of LSTM is set to be 800. In other words, each RHS is encoded into an 800-D vector by bidirectional LSTM for final classification. The dictionary size for each RNN model is about 20 MB. The optimization algorithm is the Adam [20] with mini-batch size 128. The initial learning rate is set to be 0.001. Our RNN is implemented under the Theano [3], [4] framework with either NVIDIA 6G Titan-Black or 12G Titan-X GPUs. The training of one RNN takes about 20 h. After training, the prediction/testing time (on GPU) for a single RHS is about 0.8803 ms.

C. Illustration of the Writer Identification Process

To give an intuitive understanding of our writer identification system, we show two text pages (English and Chinese) in Fig. 5(a). The blue lines are real strokes (produced in pen-down period) and the red lines are imaginary strokes (produced in pen-up period). Both the real and imaginary strokes are kept in our representation. We do not know which strokes or what kind of information are necessary and sufficient for writer identification. Fortunately, the RNN model with LSTMs can automatically learn when and how to memorize or forget a particular stroke and intelligently integrate the sequential data into a discriminative representation for final classification. In other words, our method is end-to-end by directly learning the features and classifier jointly from raw data.

Multiple short subsequences denoted as RHSs are randomly sampled from original data, as shown in Fig. 5(b). The purpose of doing so is to build a large dataset for RNN training and make ensemble-based decisions for writer identification at the page level. The extraction of RHS from raw data is just a random sampling process, which does not require any domain knowledge. Moreover, RHS is independent of the language and content of handwriting. Although character/word information is destroyed, RHS is effective for writer identification because handwriting style information is preserved.

The RNN is the core for our end-to-end writer identification system; however, it is still a black box with respect to the learned features. To reverse engineer the handwriting properties that are important for writer identification, we show in Fig. 5(c) several randomly sampled RHSs that have high individual posterior probabilities (predicted by the RNN model). These RHSs (or handwriting movements) are typical and crucial for giving high identification accuracies for the particular writer. Moreover, in our system, a small number of randomly sampled RHSs are already sufficient to guarantee the writer identification

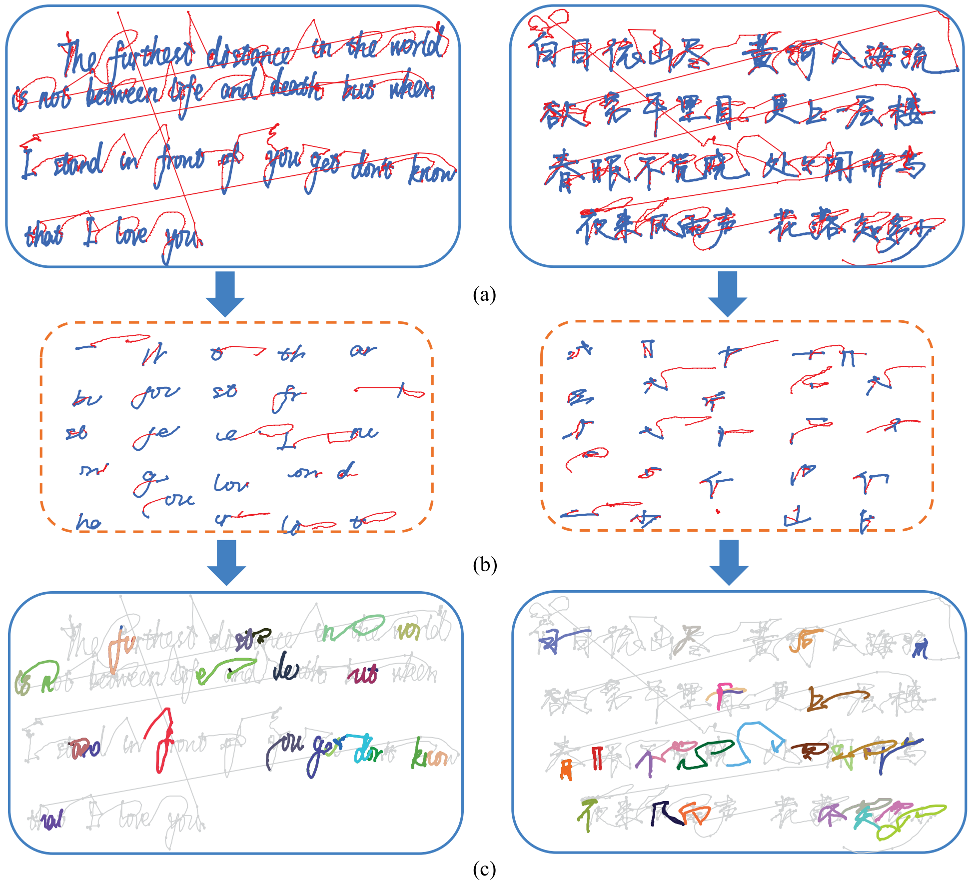

Fig. 5. (a) Illustration of the original online handwriting data in English and Chinese formats. (b) Illustration for some of the randomly sampled RHSs. In (a) and (b), each blue line is a real stroke (pen-down), and each red line is an imaginary stroke (pen-up). (c) Illustration of 30 randomly generated RHSs (represented by a randomly selected color) that have high confidences (predicted by RNN) belonging to the particular writer.

performance (see the following section), which makes the writer identification process to be both effective and efficient.

D. Varying the Number of RHSs

As shown in Fig. 5(b), a single RHS is too simple, which cannot give high accuracy for writer identification. However, by sampling many RHSs from the test data (see Fig. 4), the ensemble-based decision making can significantly boost the performance. In Fig. 6, we show the writer identification performance with respect to different number of randomly sampled RHSs. It is shown that with only one RHS, the writer identification accuracy is very low, i.e., below 50%. However, with the increasing number of RHSs, the accuracy is improved gradually and significantly. For example, with 30 RHSs, the writer identification performance on the English dataset reaches 100%. As shown in Fig. 5(c), a handwritten word may contain around five RHSs, and a sentence (let us say six words) may contain 30 RHSs. Note that the handwritten page considered in this paper (see Fig. 5) is already in a small size, i.e., around 27 words in an English text page and around 40 characters in a Chinese text page. With such a small text page, we can achieve very good writer identification performance, by the sampling of only a small number of RHSs, as shown in Fig. 5(c). These results verify the effectiveness of the proposed writer identification system.

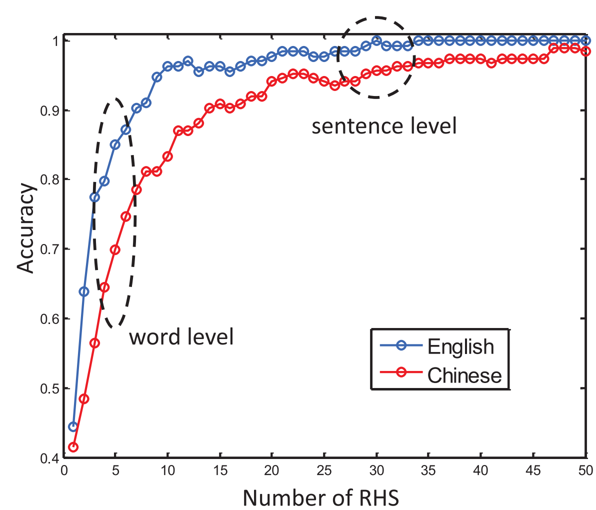

Fig. 6. Writer identification performance with different number of RHSs.

E. Comparison With Other State-of-the-Art Methods

To compare our method with other state-of-the-art approaches, Table I shows the experimental results of different methods on the BIT handwriting database [1]. All methods are compared at the page level and evaluated on the same database. The first three methods are based on hand-crafted features, while the fourth method is based on the CNN. We did not reimplement these four methods, and their accuracies can be found in [21] and [38]. As shown in Table I, our method achieves amazing performance, i.e., 100% accuracy for English and 99.46% ac-

TABLE I
COMPARISON OF THE PREDICTION ACCURACIES FOR DIFFERENT WRITER IDENTIFICATION METHODS AT THE PAGE LEVEL

	Ref.	English Dataset	Chinese Dataset	Representation	Method
Liwicki *et al.*	[24]	≈82.00%	≈80.00%	full sequence	hand-crafted features
Bulacu *et al.*	[5]	≈85.00%	≈84.00%	offline image	hand-crafted features
Li *et al.*	[21]	93.60%	91.50%	full sequence	hand-crafted features
Yang *et al.*	[38]	98.51%	95.72%	image/dropSegment	convolutional neural network
Our method	ours	**100%**	**99.46%**	sequence/RHS	recurrent neural network

curacy for Chinese writer identification, which significantly outperform the compared approaches.

The traditional hand-crafted feature-based methods cannot achieve high accuracy, because the involved feature engineering is not guaranteed to be optimal for the task of writer identification. The deep-learning-based method can automatically learn a discriminative representation from data, and thus, the CNN-based method can obtain much better accuracies. However, as shown in [38], their approach still needs some handwriting preprocessing steps and the hand-crafted features of path-signature map to train the CNN model. Contrarily, our RNN-based method is a fully end-to-end system, which does not require any domain knowledge of handwriting. The raw data of online handwriting are naturally stored as sequences (see Section II); therefore, it is straightforward and interesting to apply the RNN on the raw data for writer identification.

Previous best performance was achieved by the CNN with data augmentation strategies called dropStroke [37] and dropSegment [38], which randomly drops strokes and segments of handwritten text for writer identification. Our RHS representation shares similar idea with these approaches, namely using randomness to augment data. However, these approaches are designed for the CNN model, which operates at image level; therefore, they require some preprocessing steps such as corner detection and character segmentation [38]. Contrarily, our RHS is directly sampled at the sequence level, acting like reading a submatrix from a large matrix, as shown in Fig. 4, which makes the implementation to be very easy and efficient.

F. Discussion on Sampling Strategy

In this paper, we use the random sampling strategy, as described in Section II. Since we have sampled many RHSs from the sequence, each item (point) in the original sequence has been reused for many times. Actually, we can also use fixed-position sampling with some predefined overlap ratio to sample as many subsequences as random sampling. In our experiments, we found that fixed-position sampling can achieve the same writer identification performance compared with random sampling. However, since writer identification is a task that is independent of the content of handwriting (no need for position information) and also considering that random sampling is easier to implement, we recommend using random sampling rather than fixed-position sampling for writer identification task.

G. Effectiveness of Imaginary Stroke

As shown in Fig. 5, the imaginary (pen-up) strokes seem to be very noisy and are irrelevant to the content of handwriting; however, they may still contain some certain style information for a particular writer, which is helpful for writer identification. To verify this, we conduct experiment on the English dataset by using only the real (pen-down) strokes. In this case, the recorded imaginary strokes are ignored and replaced by virtual imaginary strokes, which are always straight lines connecting two pen-down points. To make a fair comparison, the length of the randomly sampled subsequence is also set as 100, which is the same as RHS. The comparison of the hybrid stroke and real stroke is shown in Fig. 7. When the number of random sampling is small, real stroke outperforms hybrid stroke. However, with the increasing of random sampling, the hybrid stroke becomes better than real stroke. In our experiment, by using only real strokes, we can also achieve 100% writer identification accuracy on the English dataset; however, as more as 87 random samplings are required to reach this performance. Contrarily, with hybrid strokes, we can obtain 100% accuracy with only 30 random samplings. These results suggest that the imaginary strokes also contain useful style information for writer identification.

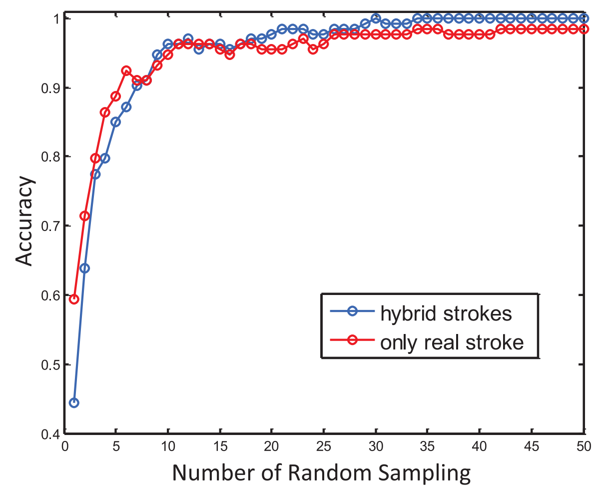

Fig. 7. Comparison of the hybrid strokes and the real strokes.

H. Cross-Language Writer Identification

Since RHS is independent of the content and language involved in handwriting, it can be used for cross-language writer identification, which is closely related to transfer learning [27]. On the third dataset (totally 133 writers with both Chinese and English text pages) from the BIT handwriting database [1], we consider the following experiments: training the RNN writer identification model using one language and testing it with another language, and training/testing with the hybrid Chinese and English handwriting contents. The experimental results are shown in Fig. 8. Since the structure of Chinese handwritten character is usually much more complex than English, the cross-language writer identification performance of "English→Chinese" is not as good as "Chinese→English." In the hybrid-language case, our method still leads to high performance. However, in the cross-language situation, the writer identification performance is not good enough. Therefore, an important future direction is to improve the transferability of the deep neural network [2].

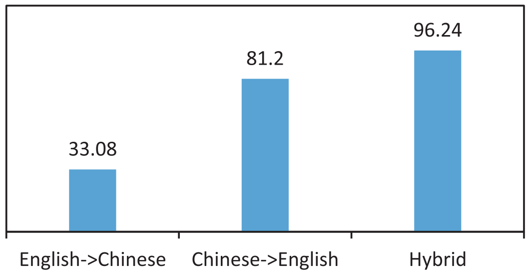

Fig. 8. Performance (%) of cross-language writer identification.

I. Future Improvement

This paper is an initial attempt on using RNN for the writer identification task. Due to the good performance, we did not try to further optimize the RNN architecture. Actually, as shown in Fig. 3, what we used is a very simple and basic RNN. In future, the methods proposed in this paper can be hopefully extended to other challenging tasks such as signature verification [14], [16], [19], [25], gender classification [23], and online script recognition [26]. In these situations, the RNN architecture used in this paper may be not powerful enough to give high performance. Therefore, further improvements should be considered, such as stacking multiple LSTM layers to build a deep network, using dropout [34] to improve the generalization performance, investigating other recurrent units such as the gated recurrent unit [7] and LSTM alternatives [18], combining recurrent and convolutional networks to better utilize spatial and temporal information, adopting the attention mechanism [6] to make model focusing on the critical part of handwriting data, and so on.

VI. Conclusion

In this paper, an end-to-end writer identification framework was proposed by using the RNN model to directly deal with the raw data of online handwriting. Multiple randomly sampled hybrid strokes were fed into a bidirectional RNN model with LSTMs for classification. Ensemble-based decision was used to boost the writer identification performance at the page level. Significant better performance was achieved by our framework compared with previous hand-crafted feature-based and CNN-based approaches. Our method is simple and reliable, which can be safely (due to the high accuracy) used in pen- and touch-based human–machine interactions. However, for the applications of user-centered systems, the interaction between users and machine is usually changed dynamically. Therefore, as discussed by Yang *et al.* [38], to design a practical and mature writer identification system, we should also consider the rejection of unknown users (e.g., by confidence analysis) and the registration of new users (e.g., by incremental or online learning).

Acknowledgment

The authors would like to thank all the reviewers for their valuable comments/suggestions and the developers of Theano [3], [4] for providing such a good deep learning platform.

References

[1] BIT handwriting database. (2007). [Online]. Available: http://biometrics.idealtest.org
[2] H. Azizpour, A. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Factors of transferability for a generic ConvNet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1790–1802, Sep. 2016.
[3] F. Bastien *et al.*, "Theano: New features and speed improvements," in *Proc. NIPS Deep Learn. Workshop*, 2012.
[4] J. Bergstra *et al.*, "Theano: A CPU and GPU math expression compiler," in *Proc. Python Sci. Comput. Conf.*, 2010.
[5] M. Bulacu and L. Schomaker, "Text-independent writer identification and verification using textural and allographic features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 701–717, Apr. 2007.
[6] K. Cho, A. Courville, and Y. Bengio, "Describing multimedia content using attention-based encoder-decoder networks," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1875–1886, Nov. 2015.
[7] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014.
[8] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.
[9] F. Gers, N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Mach. Learn. Res.*, vol. 3, pp. 115–143, 2003.
[10] A. Graves, "Generating sequences with recurrent neural networks," *arXiv:1308.0850*, 2013.
[11] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 369–376.
[12] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.
[13] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 6645–6649.
[14] D. Guru and H. Prakash, "Online signature verification and recognition: An approach based on symbolic representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1059–1073, Jun. 2009.
[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[16] D. Impedovo and G. Pirlo, "Automatic signature verification: The state of the art," *IEEE Trans. Syst., Man, Cybern.*, vol. 38, no. 5, pp. 609–635, Sep. 2008.
[17] A.K. Jain, A. A. Ross, and K. Nandakumar, *Introduction to Biometrics*. New York, NY, USA: Springer, 2011.
[18] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2342–2350.
[19] N. Kamel, S. Sayeed, and G. Ellis, "Glove-based approach to online signature verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1109–1113, Jun. 2008.
[20] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[21] B. Li, Z. Sun, and T. Tan, "Hierarchical shape primitive features for online text-independent writer identification," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 986–990.
[22] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proc. Int. Conf. Document Anal. Recognit*, 2007.
[23] M. Liwicki, A. Schlapbach, and H. Bunke, "Automatic gender detection using on-line and off-line information," *Pattern Anal. Appl.*, vol. 14, no. 1, pp. 87–92, 2011.
[24] M. Liwicki, A. Schlapbach, H. Bunke, S. Bengio, J. Mariethoz, and J. Richiardi, "Writer identification for smart meeting room systems," in *Proc. Int. Workshop Document Anal. Syst.*, 2006.
[25] M. Munich and P. Perona, "Visual identification by signature tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 200–217, Feb. 2003.
[26] A. Namboodiri and A. Jain, "Online handwritten script recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 124–130, Jan. 2004.
[27] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[28] D. Rumelhart, G. Hinton, and R. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.

[29] N. Sae-Bae and N. Memon, “Online signature verification on mobile devices,” *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 6, pp. 933–947, Jun. 2014.

[30] H. Said, T. Tan, and K. Baker, “Personal identification based on handwriting,” *Pattern Recognit.*, vol. 33, no. 1, pp. 149–160, 2000.

[31] A. Schlapbach, M. Liwicki, and H. Bunke, “A writer identification system for on-line whiteboard data,” *Pattern Recognit.*, vol. 41, no. 7, pp. 2381–2397, 2008.

[32] M. Schuster and K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[33] E. Sesa-Nogueras and M. Faundez-Zanuy, “Biometric recognition using online uppercase handwritten text,” *Pattern Recognit.*, vol. 45, no. 1, pp. 128–144, 2012.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[35] I. Sutskever, O. Vinyals, and Q. Le, “Sequence to sequence learning with neural networks,” in *Proc. Neural Inf. Process. Syst. Conf.*, 2014.

[36] K. Xu *et al.*, “Show, attend and tell: Neural image caption generation with visual attention,” in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[37] W. Yang, L. Jin, and M. Liu, “Character-level Chinese writer identification using path signature feature, dropstroke and deep CNN,” in *Proc. Int. Conf. Document Anal. Recognit.*, 2015, pp. 546–550.

[38] W. Yang, L. Jin, and M. Liu, “DeepWriterID: An end-to-end online text-independent writer identification system,” *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 45–53, Mar./Apr. 2016.