GR-RNN: Global-context residual recurrent neural networks for writer identification

Sheng He\textsuperscript{a,}\textsuperscript{*}, Lambert Schomaker\textsuperscript{b}

\textsuperscript{a} Department of Radiology, Boston Children's Hospital, Harvard Medical School, Harvard University, Boston, MA 02215, USA
\textsuperscript{b} Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, PO Box 407, Groningen 9700 AK, the Netherlands

\section{Introduction}

Forensic writer identification refers to the task of identifying a specific writer of a piece of handwriting, which has potential applications in forensic document examination \cite{1} and historical manuscript analysis \cite{2,3,4}. The classical methods \cite{5,6,7} use shape or texture features of handwritten text to recognize the writer, which requires a large amount of image information per sample in order to obtain a statistically reliable feature vector \cite{2,6}. Therefore, most studies focus on writer identification using page-level document images which contain several paragraphs or sentences. High recognition rates are reported on public datasets, such as 98.5\% \cite{9,10,11} and 98.0\% \cite{12} on the IAM \cite{13}, CVL \cite{14} and Firemaker \cite{15} datasets, respectively. However, writer identification on small amounts of data such as a word image is still a challenging problem, due to the lack of information for modeling the handwriting style of the writer. The handcrafted features completely fail in word-based writer identification, resulting in low recognition rates of 37.2\% and 30.0\% on the IAM and CVL datasets, respectively \cite{16}.

Convolutional neural networks (CNN) \cite{17,18} can extract semantic and deep abstract features by stacking several convolutional layers. This method has been widely used in pattern recognition, including writer identification \cite{11,19,20,21}. Previous studies \cite{16,21,22} show that the deep features extracted from the trained neural networks provide a good result in writer identification, even on the handwritten word images. A typical neural network \cite{18} consists of several convolutional layers which generate feature maps by the linear convolutional filters followed by a global average pooling (GAP \cite{24}). The GAP which sums out over the spatial information captures the global context while decreasing the granularity of features \cite{25}. While the ensuing generalized representation is useful in general object classification, it misses the local discriminative details which describe the sublte handwriting style information. The FragNet \cite{23} approach extracts local detailed information on fragments sampled from input images and deep feature pyramids, which can enrich the generated features and describe the handwriting style in detail. However, in the FragNet approach, the spatial context between fragments is ignored.

In fact, the global-context information and the features of local fragments are complementary to each other. Integrating those two types (i.e., levels) of information can improve the performance of writer identification based on very small amount of handwritten text. In this paper, we propose to compute the handwriting-
style information from word images, using the sequence of local fragments $x_i$ which are extracted from CNN feature maps to model the handwritten word images. To capture the context within the fragment sequence, we employ the gated recurrent unit (GRU) [26], which is a gating mechanism in recurrent neural networks (RNN), $f^t = \text{GRU}(x_i, f^{t-1})$, where $f^t$ is usually initialized to zero, to enhance the discriminative capacity of local fragment $f^t$. Inspired by the residual networks [27,28], we use the residual RNN, which is defined as $f^t = \text{GRU}(x_i, f^{t-1}) + x_i$. Thus, the feature $f^t$ of local fragment contains not only the between-fragment context information in the sequence of the fragments that are passing through the RNN, but also the original fragment feature $x_i$. In addition, instead of initializing the $f^0$ to zero, we set the $f^0$ to the global-context obtained by the global average pooling of the last layer on the neural network. Thus, the proposed method is named as global-context residual recurrent neural network (GR-RNN for short), which integrates information from the global-context and the local sequence of fragments for writer identification.

The main contribution of this paper is fourfold: 1) We propose to consider the word image as a sequence of fragments segmented on the deep feature map and apply a recurrent neural network to extract the contextual information between fragments; 2) Two kinds of complementary information from the global-context and the local fragments are integrated by the proposed global-context residual recurrent neural network (GR-RNN); 3) The proposed method is evaluated for writer identification based on word images on four public datasets and it provides better results than state-of-the-art methods; 4) Different types of images are evaluated for writer identification and we found that the performance on the gray-scale word images is better than the performance on the corresponding binarized and contour word images, indicating that the CNN focuses on the textural information on ink traces, e.g., caused by the writing instrument and varying pressure.

This paper is organized as follows: a brief summary of related work is presented in Section 2. The detailed description of the proposed GR-RNN is given in Section 3 and the experimental results is shown in Section 4. We give a conclusion in the last section.

2. Related work

This section presents a brief summary of classical methods for writer identification and the discussed approaches are not all inclusive. A comprehensive survey of writer identification can be found in [29] for interested readers. We divide methods into two groups: handcrafted features and deep learning.

2.1. Handcrafted features for writer identification

The study of writer identification can be traced back to the earliest work of Arazzi [30] in 1977, who used the histogram of the run-lengths of the background intensity as feature vectors for writer identification. After that, many different handcrafted features have been proposed for solving the forensic writer identification problem. Generally, there are three basic principles to design effective handcrafted features for writer identification.

**Principle 1:** designing textural-based features by the joint feature distribution [31]. The Hinge feature [5,6] is the joint distribution of the two angles along the handwritten contours of the ink traces. Brink et al. [3] found that the ink width takes an important role for writer identification in historical documents wrote by the Quill instrument, resulting in the Quill feature which is the joint distribution of ink direction and ink width. The oriented Basic Image Features (oBIFs) [32] which is the joint distribution of six Derivative-of-Gaussian filters at different scales is used for writer identification, with a delta encoding method. The curvature-free COLD feature [33], which is the joint distribution of the length and orientation of line fragments, is proposed for writer identification, combining the run-length of general patterns. More features generated by the joint feature distribution, such as CoHinge and Quad-Hinge, can be found in [31].

**Principle 2:** encoding the grapheme-based features can improve the performance. In [9], the SIFT descriptors with the corresponding scales and orientations are used to describe word images and a codebook is trained using the self-organizing map [5]. Christlein et al. [7] used RootSIFT descriptors and GMM supervectors as encoding method to describe the characteristic handwriting of an individual writer. Abdi et al. [34] proposed a method to synthesize graphemes based on the beta-elliptic model for Arabic writer identification. Inspired by Fraglets [6], Khalifa et al. [35] segmented connected components based on contours as graphemes which are normalized for building the codebook and the global feature descriptor. Based on the fact that junctions are prevalent in handwritten scripts, the junctures [36], which is the distribution of line width from junction points to boundary in different angles, has been used for writer identification based on a codebook trained from a set of detected junctions in handwritten images. Recently, a log path signature, called pathlets, is proposed in [10,37], which is defined as a consecutive segment on the polygonized contour and described by iterated integrals.

**Principle 3:** combining the textural and grapheme-based features can improve the performance. Schomaker and Bulacu [5] proposed to combine edge-Hinge features and the connected-component contours (CO3) on uppercase Western scripts. Later, these methods are extended to Fraglets and contour-Hinge [6] and their combination provides better performance than all individual features involved in the combination. Siddiqi and Vincent [38] proposed an automatic writer identification method by combining two types of features: the codebook-based feature captures the frequent small writing fragments of each writer and the textural feature captures the visual attributes of writing. Ahmad et al. [12] combined scale-invariant feature transform (SIFT) and RootSIFT descriptors in a set of Gaussian Mixture Models (GMM). Lai et al. [10] applied a combination of pathlet and unidirectional SIFT features for writer identification on historical document images. Aala et al. [39] used a combination of the deep and handcrafted descriptors extracted on a patch with the size of 30 × 30 from the handwritten images.

2.2. Deep learning for writer identification

Deep learning has also been widely used for writer identification. Fiel and Sablatnig [19] used the CaffeNet, which contains five convolutional layers and three fully-connected layers for deep feature extraction. A two-stream neural network with shared weights is used in [40] for writer identification based on line images in English and isolated characters in Chinese. An unsupervised deep feature learning method is proposed in [41] and the neural network is trained using the pseudo-labels generated by the cluster indices of the clustered SIFT descriptors extracted on 32 × 32 image patches. In [20], a triplet network is used to compute a similarity measure for image patches and the deep learned features are encoded using the vector of locally aggregated descriptors (VLAD) to generate the global feature vector for each document image. The conditional AutoEncoder is used in [42] to extract deep features. In [21], a semi-supervised feature learning method is proposed to learn discriminative representation and the learned deep features are encoded using VLAD for offline writer identification based on word-based images. In [22], local deep features are extracted using convolutional neural networks in character images and their sub-regions and deep features of images from the tuples are aggregated to form global features for text-independent writer identification. The AlexNet architecture with transferred knowledge
from ImageNet is employed in [43] to extract discriminating visual features from multiple representations of image patches for English and Arabic writer identification. Cilia et al. [44] evaluated several state-of-the-art deep architectures for writer identification in medieval manuscripts with a relatively small training dataset. Javidi and Jampour [45] proposed an end-to-end framework by conjugating deep learned features and a traditional handwriting descriptor for writer identification. All of these methods extract local features using deep learning and aggregate the extracted local features to form a feature vector for computing the handwriting style similarity between different individuals.

Our previous works in [16,23] train the convolutional neural networks to predict the writer identity directly from the word images. In [16], a multi-task learning strategy is applied to train the neural network with the writer identification as the main task. Deep features learned from the auxiliary tasks are adapted to the main task to improve the performance. The FragNet has been proposed in [23], which is the neural network trained on fragments extracted from both input image and the feature maps of another neural network trained for extracting feature pyramid.

2.3. Residual learning

A residual network [27] can be briefly described as: \( y = F(x, w) + x \) where \( y \) is the output, \( x \) is the input, \( w \) is the parameter and \( F \) is the neural network. With the shortcut path, the network only needs to learn the residual information \( F(x, w) = y - x \). The network structure \( F \) can be convolutional layers [27], recurrent neural networks [28] or self-attention block in transformer [46]. In our proposed method, we use the recurrent neural network (RNN) to capture the spatial context information between fragments. Unlike traditional RNN using zero values as the initial state, we use the global-context information as the initial state to integrate the global and local information.

3. Proposed method

In this section, we present the details of the proposed method for writer identification. The proposed method focuses on integrating the global-context information from the global average pooling and the sequence of fragment-based local representations segmented on deep learned feature maps.

3.1. The main architecture

As shown in Fig. 1, the network consists of four blocks and each block has two convolutional layers and one max-pooling layer. The size of the input image is \( 64 \times 128 \times 1 \) and the size of the feature maps \( f_i \) on each block is denoted as \( h \times w \times c \) where \( h \), \( w \) and \( c \) are the height, width and depth (the number of channels), respectively. The parameters of all convolutional layers are the same: kernel size is fixed to \( 3 \times 3 \), the size of stride and padding is fixed to 1. After each convolutional layer, a batch normalization layer [47] and no-linear activation function (ReLU) is applied. The max-pooling is used to reduce the spatial resolution with the kernel size of \( 2 \times 2 \) and the stride of 2. We use the global average pooling (GAP) [24] to extract the global-context \( f_g \) on the end of convolutional layers. We extract the local feature map \( f_l \) after the max-pooling layer of the third block \( f_3 \) since it contains high-level abstract feature and has a good spatial resolution. The local feature map \( f_l \) interacts with the global context \( f_g \) in the residual recurrent neural network (Residual RNN, as shown in Fig. 1) block. Finally, a fully-connected layer with softmax is used as a classifier for writer identification.

3.2. Fragment-sequence learning

Many deep learning methods use the part-based representation [25,48,49] to learn the discriminative features for recognition based on the fact that the global feature pays more attention to the global information while the part-based features segmented from the deep learned feature maps care more about the local regions of the input image. As shown in Fig. 1, the global context \( f_g \) contains global information of the handwriting style while ignores the local detailed information from fragments of the word (parts of the input image) which is very discriminative for writer identification. The fragment-based representation is complementary to the global-context information \( f_g \). Our previous work in [23] aggregates the local fragment features for writer identification and it provides a good performance. However, it simply combines the representation of fragments and the relationship between fragments has not been exploited. In this paper, we propose to use the recurrent neural network to naturally model the spatial dependency between each fragment segmented on the deep feature map \( f_l \). Instead of using zero values to initialize the hidden states of the RNN, we initialize the hidden states as the global-context \( f_g \) in order to strengthen the discriminative ability of the deep features by combining the complementary global representation and the RNN-based local representation.

In order to exploit the relationship between local fragments, the feature map \( f_l \), with the size of \( 8 \times 16 \times 256 \) (height×width×depth), is decomposed into a sequence of fragments. There are two different ways to segment the feature \( f_l \) into a sequence of fragments (as shown in Fig. 2): the horizontal segmentation which segments fragments along the height of the feature vector and the vertical segmentation which segments fragments along the width of the feature vector. As shown in Fig. 2, each feature vector describes a rectangle region which contains a fragment of handwritten strokes in the raw input word.
image given by the corresponding receptive field. The fragment segmented on the horizontal direction contains a single character, a part of character or the connection between characters, which carries the detailed handwriting style information. The fragment segmented on the vertical direction represents the layout information of the handwritten word, which also contains useful information for writer identification [50]. In this paper, the step of the horizontal segmentation is set to 1 and the step of the vertical segmentation is set to 2. Thus, the number of fragments is 8 for the two different segmentation ways and the size of each feature fragment is $1 \times 16 \times 256$ and $8 \times 2 \times 256$ for horizontal and vertical segmentation, respectively.

3.3. Global-context residual recurrent neural network

This section presents the details of the proposed residual RNN block which integrates the sequence of the fragments segmented from the local feature $f_l$ and the global-context information $f_g$. Fig. 3 shows the framework of the global-context residual RNN with the horizontal segmentation as an example.

To extract the feature vector of each fragment in the sequence, the global average pooling is used for each fragment to obtain the feature vector with the size of $1 \times 1 \times 256$. Then a fully-connected (FC) layer is used to increase the feature dimension from 256 to 512, which has the same feature dimension as the global-context feature $f_g$. Note that this FC layer is shared among all 8 fragments. Finally, a sequence of the feature vectors is obtained, denoted as $x_1, x_2, \ldots, x_N$ with the dimension of 512. The sequence of feature vectors is fed into the Gated Recurrent Units (GRU) [51] to model the spatial dependency between each fragment. The GRU is an improved version of the simple recurrent neural network that can alleviate the vanishing and exploding gradient problems and it is defined (shown in the right figure of Fig. 3) as:

$$
\begin{align*}
    z_t &= \sigma(W_x x_t + U_z f^{t-1} + b_z) \\
    r_t &= \sigma(W_x x_t + U_r f^{t-1} + b_r) \\
    h_t &= \phi(W_x x_t + U_h (r_t \odot f^{t-1}) + b_h) \\
    f^t &= z_t \odot f^{t-1} + (1 - z_t) \odot h_t
\end{align*}
$$

where $x_t$ is the fragment feature vector and $f^t$ is the global context feature at the time step $t$ ($f^0 = \tilde{f}$ for the vanilla GRU and $f^0 = f_g$ for the proposed method as shown in Fig. 1). $z_t$ and $r_t$ are the update gate and reset gate with the sigmoid $\sigma(x)$ as the activation function. $\phi(x)$ is the hyperbolic tangent function and $W_x, U_z, b_z$ are the parameters learned during training. The GRU can be denoted as $f^t = GRU(x_t, f^{t-1})$.

Note that unlike the vanilla GRU where the hidden units are initialized by zero values, the proposed method considers the global-context feature $f_g$ as the initialization of the hidden units. Thus, the proposed method is the global-context RNN and the global context feature $f_g$ is updated recurrently with the local fragment features $x_t$, $t \in \{1, 2, \ldots, 8\}$. The irrelevance information in $f^t$ is suppressed and the important information is enhanced and supplemented from the local feature $x_t$ by the two update and reset gates in GRU.

In order to fully exploit the handwriting style information in the local fragment features, we propose the global-context residual GRU, inspired by the residual network [27], as following:

$$
\hat{f}^t = GRU(x_t, f^{t-1}) + x_t \quad t \in \{1, 2, \ldots, 8\}, f^0 = f_g
$$

where the feature vector $\hat{f}^t$ is initialized by the global-context $f_g$, updated and enhanced by RNN with the local feature $x_t$ at the time step $t$. Thanks to the context modeling of GRU, each resulting feature vector $\hat{f}^t$ contains the global context and local discriminative information and thus describes the handwriting style better than its associated fragment $x_t$. Finally, the final feature vector $f$ of the input image is computed by:

$$
\hat{f} = \sum_{t=1}^{8} \hat{f}^t
$$

followed by a classifier with the softmax layer containing $N$ output neurons, corresponding to the $N$ writer identities.

3.4. Label smooth regularization

Label smooth regularization was proposed in [52] and has been used for deep feature learning [21] to mitigate the overfitting problem and to make the model more adaptable. A new ground-truth label is defined as a mixture of the original one-hot ground-truth distribution $p$ and a fixed distribution $u$:

$$
\hat{p} = (1 - \epsilon) \ast p + \epsilon u
$$

where $\epsilon$ is the smoothing parameter. Following [21, 52], we use the uniform distribution $u = 1/N$ where $N$ is the number of writers and $\epsilon = 0.1$. Finally, the training loss is defined as:

$$
\text{loss} = -(1 - \epsilon) \ast \log(p_y) - \frac{\epsilon}{N} \sum_{n=1}^{N} \log(p_n)
$$

where $y$ is the ground-truth of the writer identity. The loss not only pays attention to the ground-truth $y$ with a high weight $1 - \epsilon + \epsilon/N > 0.9$ but also takes other classes with a low weight $\epsilon/N$ into account.

4. Experimental results

4.1. Data set

We use the IAM [13], CVL [14], Firemaker [15] and CERUGEN [36] datasets to evaluate the performance of writer identification based on word images. All datasets, with the splitting of training and testing sets, are public available on the website.\(^1\)

\(^1\) https://www.ai.rug.nl/~sheng/writeridataset.html
IAM [13] contains 1,452 document images, segmenting into 82,259 word images from 657 writers. The distribution of writers on IAM is unbalanced. A half of writers contribute several pages and the rest of writers have only one page or even a few sentences. The CVL [14] dataset contains 310 writers with 96,970 word images in German and English. There are 250 and 105 writers on the Firemaker and CERUG-EN datasets, respectively. Table 1 shows the number of training and testing samples on the four datasets. Word images from one page appear only in the training or the testing set, making it possible to evaluate performance of writer identification based on line-level or page-level images. More detailed information of these datasets can be found in [23].

Table 1
The number of training and testing word images on each data set.

<table>
<thead>
<tr>
<th>Data set</th>
<th>#Writers</th>
<th>#Training</th>
<th>#Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVL [14]</td>
<td>310</td>
<td>62,406</td>
<td>34,564</td>
</tr>
<tr>
<td>Firemaker [15]</td>
<td>250</td>
<td>25,256</td>
<td>11,595</td>
</tr>
<tr>
<td>CERUG-EN [36]</td>
<td>105</td>
<td>5702</td>
<td>5127</td>
</tr>
</tbody>
</table>

4.2. Implementation details

The proposed GR-RNN model is built on the PyTorch framework. Neural networks are trained with the Adam optimizer [53]. The weight decay is set to 0.0001 and the mini-batch size is set to 16. The initial learning rate is set to 0.0001 and a decay schedule (reduce to half) at every 10 epochs is applied. The model is trained with 50 epochs. We resize all word images to a fixed size (64,128) for training the proposed neural network, similar to our previous works [16,23]. Word images are resized by keeping the aspect ratio without distortions and padding is used when it is necessary. We use the simple translation augmentation method to avoid positional bias in the data during training.

4.3. Compared models

We compare different models described in Table 2 and the main difference is how to compute the feature vector $f$ which is fed into the classifier with softmax layer for writer identification: (1) Global-context (Baseline): a classifier is built with the feature vector $f_G$ of the global-context computed by the global average pooling. This is baseline which has the similar structure with the traditional neural network without the Residual RNN block shown in Fig. 1. (2) Fragment-only (F): a classifier is built with the feature vector of the sum of the fragments segmented from the feature map $f_f$. This is similar to use the global average pooling on the feature map $f_f$. (3) Fragment-RNN (FR): applying the vanilla RNN on the sequence of the fragments with zero-initialized hidden units. (4) Fragment-Residual-RNN (FRR): using residual RNN defined in Eq. (2) on the sequence of the fragments with zero-initialized hidden units. (5) Fragment-Global-context-RNN (FGR): using the RNN on the sequence of the fragments with the global context as the initialization of the hidden units. (6) Fragment-Global-context-Residual-RNN (FGRR): the proposed global-context residual RNN model, applying the residual RNN with the global context as the initialization of the hidden units. We also compare these different networks with the horizontal and vertical fragment segmentation methods described in Fig. 1. The number of parameters and the corresponding floating point operations (FLOPs) of each model (on the IAM dataset with 657 writers) are also reported in Table 2.

4.4. Writer identification on different images

One widely accepted intuition for writer identification is that the handwriting style is encapsulated on the shape of the handwritten texts, especially on the contours of the ink traces [3,6,33,38]. Thus, the handcrafted features extracted on the contours of handwritten texts have been successfully used for writer identification based on page images. However, it has shown in [54] that convolutional neural networks are strongly biased towards recognizing objects by textures rather than shapes. In order to evaluate whether convolutional neural networks also pay attention to shapes or contours of the handwritten texts for writer identification, we use different types of input images, such as grayscale, binarized and contour images to train the neural network. Fig. 4 shows several examples of these images. Given the gray-scale image, we use the global Otsu [55] threshold to get the binarized image [7]. Then handwritten contours are extracted based on the binarized images using the tracking method proposed in [3]. All images are normalized into floating point values in the range of [0,1] and neural networks are trained with the same configuration for fair comparison.

Fig. 5 shows the performance of writer identification using different neural networks on different types of images on the four datasets. There are marked performance differences among neural networks trained with different images. CNNs trained on the grayscale images provide the best performance since the grayscale images contain rich texture information on the ink traces caused by the writing instruments and the writer’s pressures during writing.
Table 2
Different methods used to compute the feature vector $f_i$, followed by a classifier with Softmax. $f_i$ is the global context and $x_i$ is the sequence of the fragments segmented on feature map $f_i$. The number of parameters and FLOPs is computed based on the IAM dataset (with 657 writers).

<table>
<thead>
<tr>
<th>Model</th>
<th>Feature computation</th>
<th>Description</th>
<th>Parameters (MB)</th>
<th>FLOPs (G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global-context (Baseline)</td>
<td>$f = f_i$</td>
<td>The typical CNN model with the GAP</td>
<td>5.15</td>
<td>1.67</td>
</tr>
<tr>
<td>Fragment-only (F)</td>
<td>$f = \sum x_i$</td>
<td>Assemble local fragments</td>
<td>6.73</td>
<td>1.21</td>
</tr>
<tr>
<td>Fragment-RNN (FR)</td>
<td>$f = \sum f_i, f_i = \text{GRU}(x_i, f_i^{-1}), f_0 = \emptyset$</td>
<td>Vanilla RNN</td>
<td>6.73</td>
<td>1.24</td>
</tr>
<tr>
<td>Fragment-Residual-RNN (FRR)</td>
<td>$f = \sum f_i, f_i = \text{GRU}(x_i, f_i^{-1}) + x_i, f_0 = \emptyset$</td>
<td>Residual RNN</td>
<td>6.73</td>
<td>1.24</td>
</tr>
<tr>
<td>Fragment-Global-context-RNN (FGRR)</td>
<td>$f = \sum f_i, f_i = \text{GRU}(x_i, f_i^{-1}) + x_i, f_0 = f_g$</td>
<td>Vanilla RNN with global-context</td>
<td>6.73</td>
<td>1.69</td>
</tr>
<tr>
<td>Fragment-Global-context-Residual-RNN (FGR)</td>
<td>$f = \sum f_i, f_i = \text{GRU}(x_i, f_i^{-1}) + x_i, f_0 = f_{GR}$</td>
<td>Residual RNN with global-context</td>
<td>6.73</td>
<td>1.69</td>
</tr>
</tbody>
</table>

The performance on binarized images is slightly better than performance on contour images. The reason might be that binarized images contain more ink pixels than contour images, thus more valid information is kept in the neural network after several max-pooling layers.

The results indicate that the CNN tends to recognize writers according to the local textures which contain the subtle information of writer identity. In addition, with the same image, the proposed GR-RNN method provides better results than global-context neural networks (the baseline) on four datasets. Our results suggest that the texture information plays an important role, which provides a new insight for forensic examiners to consider the texture information of the ink traces for writer identification in future.

In the following sections, we conduct experiments of writer identification using the gray-scale images.

4.5. Performance of writer identification based on the gray-scale word images

In this section, we train different neural networks listed on Table 2 to evaluate the performance for writer identification with different fragment segmentation methods. Table 3 shows the performance of the different models based on the fragment sequence of vertical and horizontal segmentation for writer identification on word images. Fig. 6 shows the scatter plots of top-1 accuracy within each writer between different models. We can see that using the vanilla RNN on the sequence of fragments (FR) segmented on the low-level feature space provides the worst results. However, combining the residual information (FRR) can improve the performance for most writers (Fig. 6(c)). The performance is further boosted by integrating the global-context information (FGR and FGGR) (Fig. 6(b) and (d)). In general, models built on the
sequence of fragments with global-context (FGR, FGRR) give better results than models built on the sequence of fragments without the global-context (F, FR, FRR). In most cases, the performance of our proposed GR-RNN (FGRR) is better than baseline (Fig. 6 (a)) and other neural networks. We can also find that there is no significant difference between models using horizontal or vertical fragment segmentation methods.

4.6. Performance comparison with previous studies

Table 4 shows the comparison with the ResNet [27] and the FragNet-64 method [23]. It can be observed from the table that the proposed GR-RNN model has less parameters than FragNet-64 which has two pathways: global feature pyramid and fragment pathway. The GR-RNN has only one pathway and the sequence of fragments is segmented on the feature map. From the table we can see that our proposed methods improve the performance in terms of the Top-1 recognition rate on the four datasets.

4.7. Performance comparison with different features

We also evaluate the features \( f \) learned by the proposed GR-RNN method for writer identification, comparing with other hand-crafted features. The feature vector \( f \) is normalized by \( \hat{f} = f/\|f\|_2 \). For each writer, the model is built as the average feature of the all word images from the same writer on the training set. The nearest neighbor method with the Euclidean distance is used for writer identification given the query word images on the testing set. Table 5 shows the performance of different methods. We do not compare the traditional grapheme-based features since the information in a single word image is not sufficient to build a stable feature vector for capturing the writing style based on the bag-of-word model. This can be found in [56] that at least 150 characters are required to achieve the very satisfactory results. Generally, the deep learned features provide much better results than handcrafted features for writer identification based on word images. Our proposed GR-RNN provides the best results on the IAM, CVL and Firemaker datasets. However, it gives worse results than...
Table 6  
The writer identification performance of different neural networks using the horizontal and vertical segmentation based on line images.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>IAM</th>
<th>CVL</th>
<th>Firemaker</th>
<th>CERUG-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>94.5</td>
<td>97.4</td>
<td>99.4</td>
<td>99.9</td>
</tr>
<tr>
<td>Vertical F</td>
<td>94.3</td>
<td>97.4</td>
<td>99.4</td>
<td>99.9</td>
</tr>
<tr>
<td>FR</td>
<td>93.4</td>
<td>97.2</td>
<td>99.4</td>
<td>99.8</td>
</tr>
<tr>
<td>FRR</td>
<td>94.5</td>
<td>97.5</td>
<td>99.7</td>
<td>99.9</td>
</tr>
<tr>
<td>FGR</td>
<td>95.3</td>
<td>97.5</td>
<td>99.5</td>
<td>99.7</td>
</tr>
<tr>
<td>FGRR</td>
<td>95.2</td>
<td>97.4</td>
<td>99.5</td>
<td>99.9</td>
</tr>
<tr>
<td>Horizontal F</td>
<td>94.3</td>
<td>97.4</td>
<td>99.4</td>
<td>99.7</td>
</tr>
<tr>
<td>FR</td>
<td>93.5</td>
<td>97.2</td>
<td>99.4</td>
<td>99.7</td>
</tr>
<tr>
<td>FRR</td>
<td>94.7</td>
<td>97.3</td>
<td>99.5</td>
<td>99.7</td>
</tr>
<tr>
<td>FGR</td>
<td>94.8</td>
<td>97.4</td>
<td>99.6</td>
<td>99.9</td>
</tr>
<tr>
<td>FGRR</td>
<td>95.1</td>
<td>97.5</td>
<td>99.6</td>
<td>99.8</td>
</tr>
</tbody>
</table>

Table 7  
The writer identification performance of different neural networks using the horizontal and vertical segmentation based on page images.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>IAM</th>
<th>CVL</th>
<th>Firemaker</th>
<th>CERUG-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>95.9</td>
<td>98.2</td>
<td>99.3</td>
<td>99.6</td>
</tr>
<tr>
<td>Vertical F</td>
<td>95.9</td>
<td>98.3</td>
<td>99.1</td>
<td>99.4</td>
</tr>
<tr>
<td>FR</td>
<td>95.7</td>
<td>98.2</td>
<td>99.4</td>
<td>99.4</td>
</tr>
<tr>
<td>FRR</td>
<td>96.4</td>
<td>98.3</td>
<td>99.4</td>
<td>99.7</td>
</tr>
<tr>
<td>FGR</td>
<td>96.6</td>
<td>98.3</td>
<td>99.3</td>
<td>99.4</td>
</tr>
<tr>
<td>FGRR</td>
<td>96.4</td>
<td>98.3</td>
<td>99.3</td>
<td>99.4</td>
</tr>
<tr>
<td>Horizontal F</td>
<td>95.0</td>
<td>98.2</td>
<td>99.4</td>
<td>99.8</td>
</tr>
<tr>
<td>FR</td>
<td>95.6</td>
<td>98.2</td>
<td>99.1</td>
<td>99.4</td>
</tr>
<tr>
<td>FRR</td>
<td>96.5</td>
<td>98.2</td>
<td>99.1</td>
<td>99.4</td>
</tr>
<tr>
<td>FGR</td>
<td>96.2</td>
<td>98.3</td>
<td>99.4</td>
<td>99.6</td>
</tr>
<tr>
<td>FGRR</td>
<td>96.4</td>
<td>98.3</td>
<td>99.3</td>
<td>99.4</td>
</tr>
</tbody>
</table>

4.8. Performance on line and page images

Tables 6 and 7 show the performance of different models for writer identification based on line and page images, respectively.

Table 8  
Comparison of state-of-the-art methods on the four datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Reference</th>
<th>#Writer</th>
<th>Feature</th>
<th>Top-1 [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAM</td>
<td>Siddiqi and Vincent [38]</td>
<td>650</td>
<td>Contour and codebook features</td>
<td>91.0</td>
</tr>
<tr>
<td>He and Schomaker [31]</td>
<td>650</td>
<td>Best results among 17 handicraft features</td>
<td>93.2</td>
<td></td>
</tr>
<tr>
<td>Khalifa et al. [35]</td>
<td>650</td>
<td>Graphemes with codebook</td>
<td>92.0</td>
<td></td>
</tr>
<tr>
<td>Hadjadj and Chibani [57]</td>
<td>657</td>
<td>LPQ, RL and eBILF with OC-K-Means</td>
<td>94.5</td>
<td></td>
</tr>
<tr>
<td>Wu et al. [9]</td>
<td>657</td>
<td>Scale invariant feature transform</td>
<td>98.5</td>
<td></td>
</tr>
<tr>
<td>Lai et al. [10]</td>
<td>650</td>
<td>Pathlet + bVLAD</td>
<td>98.5</td>
<td></td>
</tr>
<tr>
<td>Khant et al. [12]</td>
<td>650</td>
<td>SIFT + RootSIFT</td>
<td>97.8</td>
<td></td>
</tr>
<tr>
<td>Nguyen et al. [22]</td>
<td>650</td>
<td>CNN with sub-images of size 64 x 64</td>
<td>93.1</td>
<td></td>
</tr>
<tr>
<td>FragNet-64 [23]</td>
<td>657</td>
<td>CNN with word images and fragments</td>
<td>96.3</td>
<td></td>
</tr>
<tr>
<td>Proposed</td>
<td>657</td>
<td>Global-context recurrent neural network</td>
<td>96.4</td>
<td></td>
</tr>
<tr>
<td>CVL</td>
<td>Fiel and Sablatnig [58]</td>
<td>309</td>
<td>SIFT with GMM</td>
<td>97.8</td>
</tr>
<tr>
<td>Lai et al. [10]</td>
<td>310</td>
<td>Pathlet + bVLAD</td>
<td>99.76</td>
<td></td>
</tr>
<tr>
<td>Christlein et al. [7]</td>
<td>310</td>
<td>SIFT with GMM and Exemplar-SVMs</td>
<td>99.2</td>
<td></td>
</tr>
<tr>
<td>Khan et al. [12]</td>
<td>310</td>
<td>SIFT + RootSIFT</td>
<td>99.0</td>
<td></td>
</tr>
<tr>
<td>Chen et al. [21]</td>
<td>310</td>
<td>Semi-supervised with ResNet-50 trained with IAM</td>
<td>99.2</td>
<td></td>
</tr>
<tr>
<td>FragNet-64 [23]</td>
<td>310</td>
<td>CNN with word images and fragments</td>
<td>99.1</td>
<td></td>
</tr>
<tr>
<td>Proposed</td>
<td>310</td>
<td>Global-context recurrent neural network</td>
<td>99.3</td>
<td></td>
</tr>
<tr>
<td>Firemaker</td>
<td>He and Schomaker [31]</td>
<td>250</td>
<td>Best results among 17 handicraft features</td>
<td>92.2</td>
</tr>
<tr>
<td>Wu et al. [9]</td>
<td>250</td>
<td>Scale invariant feature transform</td>
<td>92.4</td>
<td></td>
</tr>
<tr>
<td>Khan et al. [12]</td>
<td>250</td>
<td>SIFT + RootSIFT</td>
<td>98.0</td>
<td></td>
</tr>
<tr>
<td>Nguyen et al. [22]</td>
<td>250</td>
<td>CNN with sub-images of size 64 x 64</td>
<td>93.6</td>
<td></td>
</tr>
<tr>
<td>Lai et al. [10]</td>
<td>250</td>
<td>Pathlet + bVLAD</td>
<td>97.98</td>
<td></td>
</tr>
<tr>
<td>FragNet-64 [23]</td>
<td>250</td>
<td>CNN with word images and fragments</td>
<td>97.6</td>
<td></td>
</tr>
<tr>
<td>Proposed</td>
<td>250</td>
<td>Global-context recurrent neural network</td>
<td>98.8</td>
<td></td>
</tr>
<tr>
<td>CERUG</td>
<td>He and Schomaker [31]</td>
<td>105</td>
<td>Best results among 17 handicraft features</td>
<td>97.1</td>
</tr>
<tr>
<td>FragNet-64 [23]</td>
<td>105</td>
<td>CNN with word images and fragments</td>
<td>100.0</td>
<td></td>
</tr>
<tr>
<td>Proposed</td>
<td>105</td>
<td>Global-context recurrent neural network</td>
<td>99.1</td>
<td></td>
</tr>
</tbody>
</table>
contains several word images. In addition, there is no significant difference among the performance of different neural networks on page level (as shown in Table 7). Thus, given a large amount of training data, the basic neural network with several convolutional layers can provide satisfactory performance.

4.9. Performance comparison with state-of-the-art methods

Table 8 presents the comparison with state-of-the-art methods for writer identification based on page images. Although different methods in this table may use different experimental protocols, it is still interesting to draw some conclusions. For IAM dataset, the traditional handcrafted feature [9] provides the best results than the deep learning methods. The possible reason might be that the number of samples for each writer is unbalanced. Some writers contributed several pages and some only wrote one page with several sentences. Our proposed method provides better results than deep learning methods [22,23] on the IAM dataset. The CVL and CERUG-EN datasets are the easy datasets for writer identification and our proposed method provides a comparative results with other studies. The proposed method provides the best performance on the Firemarker dataset.

5. Conclusion

In this paper, we have introduced a novel global-context residual recurrent neural network framework for writer identification, which jointly employs the global context and the local fragment-part information. The advantage of the proposed method is that it can capture the global-context information extracted by the convolutional network and local fine-grained information extracted by the recurrent neural network. Thus, it can extract the detailed writing style information from one single word image and it achieves state-of-the-art performance on four public datasets.

The limitation of the proposed method is that it needs the word or sub-word image segmentation, which requires extra prepossessing steps for applying it on other documents (such as historical document fragments [59]). In future work, the proposed method can be extended to capture the handwriting style information on any handwritten document without segmentation by using a sliding window strategy.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[40] L. King, Y. Qiao, Deepwriter: a multi-stream deep CNN for text-independent


[57] B. Hadjadi, Y. Chibani, Two combination stages of clustered one-class classifiers for writer identification from text fragments, Pattern Recognit. 82 (2018) 147–162.


Sheng He gained a cum laude Ph.D. degree in artificial intelligence from the University of Groningen, the Netherlands, in 2017. From 2017 to 2018, he was a Post-doctoral fellow at the University of Groningen. In 2018, he joined Harvard Medical School as a research fellow. He received the Chinese government award for outstanding self-financed students abroad (2016) from the Chinese Scholarship Council and the Charles A. King Trust Postdoctoral Research Fellowship (2020, in Massachusetts, USA). His research interests include handwritten document analysis, deep learning, and medical image analysis.

Lambert Schomaker is Professor in Artificial Intelligence at the University of Groningen and was the director of its AI institute ALICE from 2001 to 2018. His main interests are pattern recognition and machine learning problems, with applications in handwriting recognition problems. He has contributed to over 200 peer-reviewed publications in journals and books (h=17/ISI, h=42/Google Citations). His work is cited in 23 patents. In recent years, his focus has been on continuous-learning systems and bootstrapping problems, where learning starts using very few examples. Prof. Schomaker is a senior member of IEEE, a member of the IAPR, and a member of Dutch research program committees in e-Science (at the NWO), Computational Humanities (at the KNAW), Computational science, and energy (at Shell/NWO/FOM). He received two IBM Faculty Awards (in 2011 and 2012) for the Monk word-retrieval system in historical manuscript collections using high-performance computing.