Pattern Recognition 103 (2020) 107305

Contents lists available at ScienceDirect

Pattern Recognition

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

Radical analysis network for learning hierarchies of Chinese characters

Jianshu Zhang, Jun Du*, Lirong Dai

National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP), University of Science and Technology of China, No. 96, JinZhai Road, Hefei, Anhui P. R. China

ARTICLE INFO

Article history:
Received 12 July 2019
Revised 12 February 2020
Accepted 23 February 2020
Available online 28 February 2020

Keywords:
Radical
Attention
Chinese character
Few-/zero-shot learning

ABSTRACT

Chinese characters have a valuable property, this is, numerous Chinese characters are composed of a compact set of fundamental and structural radicals. This paper introduces a radical analysis network (RAN) that makes full use of this valuable property to implement radical-based Chinese character recognition. The proposed RAN employs an attention mechanism to extract radicals from Chinese characters and to detect spatial structures among the radicals. Then, the decoder in RAN generates a hierarchical composition of Chinese characters based on the knowledge of the extracted radicals and their internal structures. The method of treating a Chinese character as a composition of radicals rather than as a single character category is a human-like method that can reduce the size of the vocabulary, ignore redundant information among similar characters and enable the system to recognize unseen Chinese character categories, i.e., zero-shot learning. Through experiments, we assess the practicality of RAN for recognizing Chinese characters in natural scenes. Furthermore, a RAN framework can be proposed for scene text recognition with the extension of a dense recurrent neural network (denseRNN) encoder, a multihead coverage attention model and HSV representations. The proposed approach achieved the best performance in the ICPR MTWI 2018 competition.

1. Introduction

Automatic recognition of Chinese text has considerable commercial value and social benefits, as Chinese is one of the most widely adopted reading systems in the world: nearly one-quarter of the world's population reads and writes in Chinese scripts.

However, the recognition of Chinese characters or texts is among the most challenging topics in pattern recognition [1]. Although recent deep-learning-based approaches [2], such as the convolutional neural network (CNN) [3,4] and recurrent neural network (RNN) [5,6], have achieved considerable success in the recognition of approximately 4,000 commonly used Chinese characters [7,8], there are many uncommonly used Chinese characters in addition to these 4,000 common characters. The numerous character categories (more than 25,000), complex internal structures and confusion among similar characters result in difficulties for recognizing both commonly used and uncommonly used Chinese characters.

Additionally, recognition of rarely used Chinese characters is typically a few-shot learning problem since samples of such character categories are difficult to collect. Moreover, recognition of some novel Chinese characters (e.g., the character "Duang", newly created by Jackie Chan, see Fig. 9) is a zero-shot learning problem, as these characters are newly created and have never been seen before. Few-/zero-shot learning has attracted the interest of researchers [9,10]. This type of learning is a challenging problem but has enormous potential value, as the ability to learn and generalize from a few examples is a hallmark of human intelligence [11] that is difficult to achieve with deep learning methods.

In this paper, we propose a novel deep-learning-based model, called the radical analysis network (RAN), for Chinese character recognition. The proposed RAN makes full use of the intensely hierarchical nature of Chinese characters. Unlike English or Arabic characters, Chinese characters are composed of basic components [12] (called radicals in this paper). A small number of radicals can be used to construct many Chinese characters [13]. Therefore, an intuitive method for Chinese character recognition is to decompose Chinese characters into radicals and analyze the hierarchical structures among the radicals.

In RAN, we employ an attention mechanism [14] to implicitly perform radical extraction and structure analysis. By utilizing an attention model, RAN can focus on certain components of a Chinese character and choose the most relevant component to describe a radical. Meanwhile, we prove that the attention model can also detect the relative spatial relationships among radicals. Based on the knowledge of detected radicals and spatial relation-

* Corresponding author.
E-mail addresses: xysszjs@mail.ustc.edu.cn (J. Zhang), jundu@ustc.edu.cn (J. Du), lrdai@ustc.edu.cn (L. Dai).

https://doi.org/10.1016/j.patcog.2020.107305
0031-3203/

ships, RAN employs a decoder to analyze the internal hierarchical radical structures of Chinese characters. Finally, we can map these hierarchical radical structures to their corresponding character categories for recognition. Compared with traditional character-based methods, RAN has two distinct properties: (i) the size of the radical vocabulary is largely reduced compared to that of the character vocabulary; and (ii) RAN is a novel zero-shot learning technique for Chinese character recognition. Unseen Chinese characters can be recognized because the necessary radicals and structures have been learned from characters observed during training.

In addition to the Chinese character recognition task, this paper investigates the application of RAN in Chinese text line recognition. An advantage of RAN is that can be extended from recognizing isolated characters to recognizing text lines, as the embedded attention mechanism, which was originally devised to extract radicals, is also able to detect Chinese characters in text lines.

To adapt RAN to the text line problem and to capture the document's temporal layout, we incorporate a new source encoder layer in the form of a multirow bidirectional RNN combined with gated recurrent units (GRUs) [6] before the application of attention, and the GRU layers are improved by employing rectified linear unit (ReLU) activation and batch normalization layers [15]. We also improve our attention model from single-head coverage attention to multihead coverage attention, where each head can generate a different attention distribution, to improve performance. This improvement enables the decoder to simultaneously focus on context radicals and structures in each decoding step. Furthermore, we combine HSV channels with RGB channels to strengthen RAN's robustness [16] on text line images containing complex and difficult noisy backgrounds. Evaluated on the MTWI-18 [17] and RCTW-17 [18] text line benchmarks, which are both dominated by Chinese characters, with many low-frequency or even unseen Chinese characters, the proposed RAN has fully leveraged its ability of few-shot or even zero-shot learning and shown its distinct advantages.

The main contributions of this study are as follows:

- We propose RAN, a novel radical-based Chinese character processing method with few-/zero-shot learning ability.
- We describe the hierarchical radical structure of 27,533 Chinese characters (all the Chinese characters in the GB18030 standard [19]) and release the results to benefit related research.
- We demonstrate the performance of RAN on an unseen Chinese character recognition task and compare RAN with character-based methods on a seen Chinese character recognition task.
- We introduce a RAN extension for text line recognition and experimentally demonstrate its performance.

This paper is an extension of our previous conference paper [20] in five ways: 1) We exploit densely connected convolutional networks (DenseNet) [21] in RAN; 2) We modify the composition of Chinese characters in ideographic description sequence (IDS) format; 3) We evaluate RAN on a natural scene database and prove its practical value in the real world; 4) We compare RAN with character-based methods and provide a detailed experimental analysis; and 5) We extend RAN for text line recognition and analyze its performance in detail. Moreover, we have also evaluated RAN on handwritten Chinese character recognition problems [22,23], but we did not investigate its application on text line recognition. The convincing results on handwritten Chinese character recognition further proved the practical value and generalizability of RAN.

The rest of this paper is organized as follows: Section 3 introduces the hierarchical radical structure of Chinese characters. Section 4 describes the proposed framework of RAN. Section 5 presents the architecture of the extension of RAN for Chinese text line recognition. Section 6 introduces the implementation of the training and testing procedures. Section 7 reports the experimental results on Chinese character recognition. Section 8 reports the experimental results on Chinese text line recognition, and Section 9 presents concluding remarks.

2. Related work

2.1. Character-based Chinese character recognition

Traditional character-based methods treat Chinese character recognition problems as classification problems. For example, in offline Chinese character recognition, offline characters are naturally represented as scanned images; therefore, the CNN is a natural and effective method for offline Chinese character recognition [24–26], as the strong a priori knowledge of convolution makes the CNN a powerful model for image classification. With respect to online recognition, pen movements (xy-coordinates) are stored as sequential data, which can be naturally processed by using the RNN [27]. Moreover, the CNN can be applied to online characters by first transforming the online handwriting trajectory into image-like representations such as AMAP[28], path signature maps [29] and directional feature maps [30].

2.2. Radical-based Chinese character recognition

The use of radicals for Chinese character recognition has been researched for decades, and radical-based Chinese character recognition can be considered to consist of two major problems, namely, radical extraction [31,32] and structure analysis [33,34]. The two problems can be solved sequentially or globally.

2.2.1. Sequential methods

Most conventional radical-based approaches are sequential approaches in which radical extraction is the first stage of a two-stage Chinese character recognition process. The second stage requires analysis of the structures of the radicals to identify the optimal radical combination. For example, [35] first implemented radical extraction based on a nonlinear active shape modeling (ARM) method. During the structural analysis, a dynamic tunneling algorithm was used to search for the optimal shape parameters in terms of chamfer distance minimization. Finally, the complete characters can be recognized via the Viterbi algorithm. Additionally, in [36], a recursive hierarchical scheme is developed to first perform radical extraction. Character features and radical features are then extracted for matching. Finally, in the structure analysis stage, a hierarchical radical matching scheme is devised to identify the radicals embedded in an input Chinese character and to recognize the input character.

2.2.2. Global methods

Different from conventional sequential methods, the RAN proposed in this study is a global method. RAN aims to address the following limitations of sequential methods: (i) radical extraction is a difficult problem; and (ii) structure analysis is complex, and an effective strategy [37] must be applied during radical combination. Other radical-based Chinese character recognition methods are also performed in a global manner. A multilabel learning with a residual network architecture was proposed in [38]. The method first predefined twenty types of radical structures and then marked every radical with a specific position. The Chinese characters are recognized when the labeled position-dependent radicals are predicted successfully. [39] proposed oversegmentation of the Chinese character graph into candidate radicals to avoid radical extraction. The optimal radical segmentation is searched in a lexicon-driven manner using a beam search strategy. However, the proposed method can address only the left-right structure, which is the simplest structure of Chinese characters.

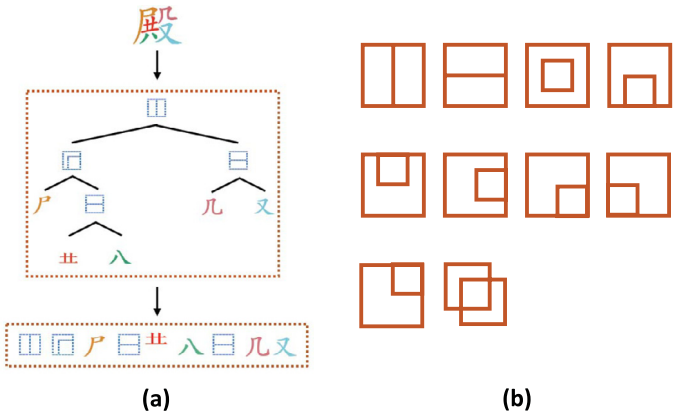


Fig. 1. (a) Hierarchical radical structure of an example Chinese character. The radicals are on the leaf nodes, and the structures are at the parent nodes. (b) Graphical representation of 10 common radical structures.

2.3. Character-based Chinese text recognition

Both handwritten Chinese text recognition and scene text recognition are active research problems. For handwritten text recognition, [40] proposed a hybrid method using the hidden Markov model (HMM) to avoid segmentation problems. [41] proposed a multi-spatial-context fully convolutional neural network followed by a residual recurrent network to avoid segmentation problems. The path signature representation was exploited, and an implicit language model was developed to complete sequence prediction. For scene text recognition, there are few benchmarks specifically for Chinese text line recognition. However, the approaches proposed for English text recognition [42–46] are popular in Chinese text recognition. Both [42] and [43] proposed end-to-end trainable systems by employing connectionist temporal classification (CTC) for decoding, while [42] proposed the convolutional recurrent neural network (CRNN) architecture for encoding the sequential layout, and [43] proposed a sliding convolution character model. Both [44] and [45] employed the encoder-decoder framework for end-to-end training, while [44] combined a rectification network to effectively deal with irregular text lines, and [45] proposed a focusing attention (FA) model to help improve the alignment of attention. [46] proposed a method named aggregation cross-entropy (ACE) for accelerating the training of sequence recognition.

The above methods focused on character-based text recognition. However, in this paper, we investigated a valuable ability of the proposed RAN that has not been investigated in previous radical-based Chinese character recognition methods: radical-based Chinese text recognition.

3. Radical analysis

Unlike English or Arabic characters, many Chinese characters can be decomposed into a limited number of radicals. These radicals are viewed as semantic parts shared by different characters that appear in specific positions.

3.1. Radical structures

Some radical structures can be derived based on position-dependent radicals. For example, a left-hand radical and a right-hand radical constitute a left-to-right structure (LR structure in Fig. 1(b)). Ten different Chinese radical structures are predefined (Chapter 12 in [3]), and we list them in Fig. 1(b): (1) left-to-right structure (LR), (2) above-to-below structure (AB), (3) full-surround structure (S), (4) surround-from-above structure (SA), (5) surround-from-below structure (SB), (6) surround-from-left structure (SL), (7) surround-from-upper-left structure (SUL), (8) surround-from-upper-right structure (SUR), (9) surround-from-lower-left structure (SLL), and (10) overlaid structure (O).

3.2. Hierarchical radical structures

The internal radical structures of Chinese characters are intensely hierarchical: each Chinese character is first composed of a main structure; then, the main structure will be decomposed into several substructures until the final structures cannot be further decomposed. We illustrate the hierarchical radical structure of a Chinese character as a tree in Fig. 1(a). The Chinese character instance is above the top of the tree, and different radicals are denoted with different colors. The following tree structure describes the hierarchical radical structure: symbols on the parent nodes denote radical structures, while symbols on the leaf nodes denote radicals. The main structure of the instance character is the left-to-right structure (top of the tree). The left part of the tree indicates a surround-from-upper-left structure, and the right part of the tree indicates an above-to-below structure. Finally, an above-to-below structure is presented at the bottom-right part of the surround-from-upper-left structure. As for those Chinese characters which are also radicals, they have no further internal structures, therefore their hierarchical radical structures are just themselves.

3.3. Ideographic description sequence (IDS)

The bottom of Fig. 1(a) shows the IDS sequence of the example Chinese character, which is converted from the hierarchical tree structure by following a depth-first traversal order. We decompose Chinese characters into IDS sequences using the strategy in cjkvi-ids[1]. All 27,533 Chinese characters in the GB18030 standard can be decomposed into 485 radicals and 10 radical structures in IDS format, which is a more compact and more reasonable representation than the one introduced in our previous paper [20] as we omit many strange radical structures. Fig. 2 shows the statistical information about the decomposition of the 27,533 Chinese characters, and Fig. 2(a) illustrates how many Chinese characters are related to each radical structure. The left-to-right structure (LR) and above-to-below structure (AB) dominate the Chinese character set, as they are basic and common structures. Fig. 2(b) illustrates how many Chinese characters are related to each radical. For brevity, we show only the 8 most common radicals on the left. Some low-frequency radicals are included in the complete set, which introduces difficulties for RAN. We present 5 examples that appear only once in the radical set on the right. These low-frequency radicals are usually related to rarely used Chinese characters. Our generated IDS sequence of the 27,533 Chinese characters is publicly available[2].

4. Radical analysis network

As illustrated in Fig. 3, by considering Chinese characters as linearized hierarchical radical structures (IDS sequences), the proposed RAN can successfully recognize a Chinese character by first predicting its IDS sequence and then selecting the output character category by searching in the predefined IDS dictionary to find the character category whose IDS sequence is most like the predicted IDS sequence. The IDS dictionary links the 27,533 Chinese characters with specific IDS sequences. If the input image belongs

[1] https://github.com/cjkvi/cjkvi-ids.
[2] https://github.com/JianshuZhang.

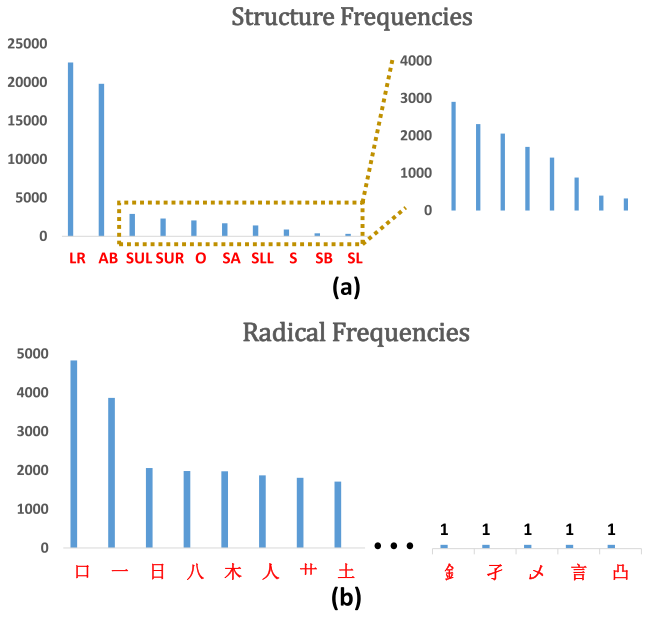


Fig. 2. (a) Number of Chinese characters with a specific structure; (b) Number of Chinese characters with a specific radical.

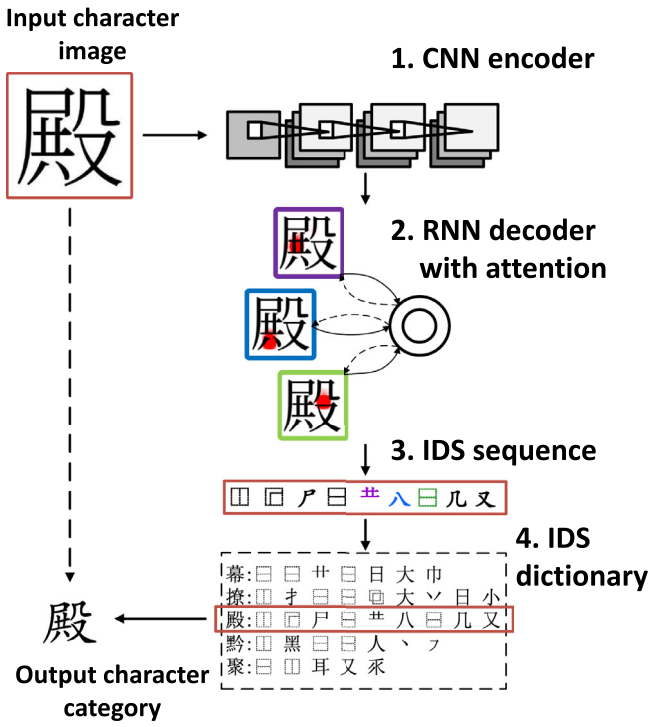


Fig. 3. Overall architecture of RAN: 1. The input character image is first fed into a CNN encoder to be transformed into visual features; 2. An RNN equipped with an attention model is employed as a decoder, the attention model lets the decoder focus on useful parts of the visual features; 3. An IDS sequence is then predicted by the decoder symbol by symbol; 4. RAN finally outputs a character category by searching in the IDS dictionary to find a character category whose IDS sequence is most like the predicted IDS sequence.

to a character category that is not observed during training but is included in the IDS dictionary, RAN recognizes it in the same manner as that for observed character categories. If the input image is a newly created character category that is not included in the current IDS dictionary, RAN can still predict the IDS sequence. All we need to do is update the IDS dictionary; there is no need to collect training samples of that category or retrain the models.

Regarding the network architecture, RAN is an improved version of the attention-based encoder-decoder framework. [47] recently showed that a caption sequence can be generated from an image with an attention-based encoder-decoder framework. The attention-based encoder-decoder framework was first proposed in [48] for machine translation. This framework has been extensively applied to many other applications, including speech recognition [49,50], image captioning [51,52] and formula recognition [53–55].

4.1. Dense encoder

As shown in Fig. 3, RAN consists of an encoder and a decoder. Depending on the a priori knowledge of convolution, the CNN has proven to be a powerful model for image processing. Therefore, we first employ a CNN as the encoder to convert input character images to high-level visual features. Moreover, the convolutional layers in the CNN encoder are configured as densely connected layers in DenseNet [21].

The main idea of DenseNet is to use the concatenation of the output feature maps of the preceding layers as the input of the succeeding layers. As DenseNet is composed of many convolutional layers, let $H_l(\cdot)$ denote the convolution function of the l^{th} layer; then, the output of layer l is represented as:

$$\mathbf{a}_l = H_l([\mathbf{a}_0; \mathbf{a}_1; \ldots; \mathbf{a}_{l-1}]) \tag{1}$$

where $\mathbf{a}_0, \mathbf{a}_1, \ldots,$ and $\mathbf{a}_l$ denote the output features produced in layers $0, 1, \ldots,$ and l, and "; " denotes the concatenation operation of feature maps, the number of output channels of each convolutional layer H_l is called growth rate k. This iterative connection enables the network to learn shorter interactions across different layers and reuse features computed in the preceding layers. In this manner, DenseNet strengthens feature extraction and facilitates gradient propagation.

An essential component of convolutional networks is the pooling layers, which can increase the receptive field and improve invariance. However, the pooling layers disenable the concatenation operation as the size of the feature maps changes. Additionally, DenseNet is inherently memory demanding because the number of interlayer connections increases quadratically with depth. Consequently, DenseNet is divided into multiple densely connected blocks, and we employ compression layers between two contiguous dense blocks to reduce memory consumption. We illustrate the detailed architecture of the proposed dense encoder in Fig. 6 in Section 6.1. Rather than extracting features after a fully connected layer, the dense encoder contains only convolutional, pooling and activation layers, acting as a fully convolutional neural network, which enables the subsequent decoder to selectively focus on certain pixels of an image by choosing specific portions from the extracted visual features.

We introduce the high-level visual features extracted by the dense encoder as $\mathbf{A}$, which is a three-dimensional array of size $H \times W \times C$, where H denotes the height, W denotes the width and C denotes the output channels. Therefore, the features $\mathbf{A}$ can be seen as a grid of $H \times W$ elements, where each element is a C-dimensional vector corresponding to a local region of the image: $\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_{H\times W}\}, \mathbf{a}_i \in \mathbb{R}^C$.

4.2. Decoder with attention

After extracting the visual features from the input images, the decoder of RAN begins to generate the IDS sequence. The IDS sequence is denoted as $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$, $\mathbf{y}_i \in \mathbb{R}^K$, where K is the number of symbols in the radical vocabulary, which includes 485

basic radicals and 10 spatial structures, and T is the length of the IDS sequence. Note that the length of the IDS sequence is variable; we have to predict the output sequence one symbol at a time.

Intuitively, the entire input image is not required to predict each radical or structure; only related pixels need to contribute. Therefore, we employ an attention mechanism to address the problem of alignment and to let the decoder know which part of the input image is suitable for generating the next predicted radical or structure. For example, in Fig. 3, the purple, blue and green rectangles denote three symbols, with the red color representing the attention probabilities of each radical or structure (a darker red color denotes a higher probability). When predicting the above-to-below structure (green rectangle), the attention model can automatically focus on the area between two vertical radicals, indicating an above-to-below direction, and the alignment of the radicals corresponds to human intuition.

4.2.1. Gated recurrent units (GRUs)

We employ a GRU [56], an improved version of the simple RNN that can alleviate the vanishing and exploding gradient problems, as the decoder. The GRU decoder is also implemented with the batch normalization function. Given the input $\mathbf{x}_t$, the GRU output $\mathbf{h}_t$ is computed by:

$$\mathbf{h}_t = \mathrm{GRU}(\mathbf{x}_t, \mathbf{h}_{t-1}) \tag{2}$$

and the GRU function can be expanded as follows:

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_{xz}\mathrm{BN}(\mathbf{x}_t) + \mathbf{U}_{hz}\mathbf{h}_{t-1}\right) \tag{3}$$

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_{xr}\mathrm{BN}(\mathbf{x}_t) + \mathbf{U}_{hr}\mathbf{h}_{t-1}\right) \tag{4}$$

$$\tilde{\mathbf{h}}_t = \mathrm{ReLU}\left(\mathbf{W}_{xh}\mathrm{BN}(\mathbf{x}_t) + \mathbf{U}_{rh}(\mathbf{r}_t \otimes \mathbf{h}_{t-1})\right) \tag{5}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \otimes \mathbf{h}_{t-1} + \mathbf{z}_t \otimes \tilde{\mathbf{h}}_t \tag{6}$$

where σ is the sigmoid function, BN is the batch normalization function and $\otimes$ is an elementwise multiplication operator. $\mathbf{z}_t$, $\mathbf{r}_t$ and $\tilde{\mathbf{h}}_t$ are the update gate, reset gate and candidate activation, respectively. For brevity, we use GRU to represent the GRU layer in Eq. (2) and do not expand it.

4.2.2. First GRU layer

The attention-based decoder adopts two unidirectional GRU layers and an attention model to learn an alignment between the output symbol $\mathbf{y}_t$ and the input image $\mathbf{X}$ in each decoding time step t. Since we have converted the input image into high-level visual features through a dense encoder, the attention model is employed to learn the alignment between the output symbol $\mathbf{y}_t$ and the features $\mathbf{A}$. Let $\mathbf{s}_t$ denote the output state of the decoder at time step t. Since we do not have $\mathbf{y}_t$ when we want to compute the attention probabilities between $\mathbf{y}_t$ and $\mathbf{A}$, standard attention mechanisms replace $\mathbf{y}_t$ with the previous decoder state $\mathbf{s}_{t-1}$ to compute the attention probabilities. By contrast, in this paper, we utilize $\hat{\mathbf{s}}_t$ rather than $\mathbf{s}_{t-1}$ to compute the attention probabilities because we believe that $\mathbf{s}_{t-1}$, the decoder state of the previous step, is an inaccurate representation of the current alignment information. We call $\hat{\mathbf{s}}_t$ the prediction of the current GRU hidden state, which is computed by the previous ground-truth symbol $\mathbf{y}_{t-1}$ and the previous decoder state $\mathbf{s}_{t-1}$:

$$\hat{\mathbf{s}}_t = \mathrm{GRU}(\mathbf{y}_{t-1}, \mathbf{s}_{t-1}) \tag{7}$$

4.2.3. Coverage-based attention

Before utilizing $\hat{\mathbf{s}}_t$ and $\mathbf{A}$ to compute the attention probabilities, we must introduce a coverage vector $\mathbf{F}$, which is computed based on the summation of all past attention probabilities. The vector is computed by feeding the past attention probabilities into a convolutional layer:

$$\mathbf{F} = \mathbf{Q} * \sum_{j=1}^{t-1} \boldsymbol{\alpha}_j \tag{8}$$

Here, $\mathbf{Q}$ denotes a convolution layer, and $\boldsymbol{\alpha}_j$ denotes the attention probabilities at decoding step j. The coverage vector $\mathbf{F}$ is employed to address the difficulty of the standard attention mechanisms, namely, the lack of coverage [57], which usually leads to problems with over-parsing (some radicals are decoded more than once) and under-parsing (some radicals are never decoded). The past alignment information contained in $\mathbf{F}$ helps the attention model know which part of the input image has been attended and ensures that each part is attended once and only once. We initialize $\mathbf{F}$ as a zero vector. Then, we compute the energy coefficients between $\hat{\mathbf{s}}_t$ and $\mathbf{A}$ given $\mathbf{F}$ using the following multilayer perceptron:

$$e_{ti} = \boldsymbol{\nu}_{\mathrm{att}}^{\mathrm{T}} \tanh(\mathbf{W}_{\mathrm{att}}\hat{\mathbf{s}}_t + \mathbf{U}_{\mathrm{att}}\mathbf{a}_i + \mathbf{U}_f\mathbf{f}_i) \tag{9}$$

Here, e_{ti} denotes the energy of feature vector $\mathbf{a}_i$ (elements of $\mathbf{A}$) in decoding step t, and $\mathbf{f}_i$ denotes the elements of $\mathbf{F}$. Let n and n' denote the dimensions of GRU decoder and attention, q denotes the number of output channels of convolution layer Q, $\boldsymbol{\nu}_{\mathrm{att}} \in \mathbb{R}^{n'}$, $\mathbf{W}_{\mathrm{att}} \in \mathbb{R}^{n' \times n}$, $\mathbf{U}_{\mathrm{att}} \in \mathbb{R}^{n' \times C}$, $\mathbf{U}_f \in \mathbb{R}^{n' \times q}$. We can obtain the attention coefficients α_{ti} by feeding e_{ti} into a softmax function. A context vector $\mathbf{c}_t$ is computed by the weighted summation of all feature vectors. We call $\mathbf{c}_t$ the context vector since it contains the overall information of the input image. However, as the weights α_{ti} denote the alignment probabilities, $\mathbf{c}_t$ includes the information of only the useful part of the image rather than the entire image:

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{H \times W} \exp(e_{tk})} \qquad \mathbf{c}_t = \sum_{i=1}^{H \times W} \alpha_{ti}\mathbf{a}_i \tag{10}$$

4.2.4. Second GRU layer

We finally utilize the second GRU layer to calculate the current output state of decoder $\mathbf{s}_t$ given $\mathbf{c}_t$ and $\hat{\mathbf{s}}_t$:

$$\mathbf{s}_t = \mathrm{GRU}\left(\mathbf{c}_t, \hat{\mathbf{s}}_t\right) \tag{11}$$

The probability of each predicted symbol is computed by the context vector $\mathbf{c}_t$, the current GRU hidden state $\mathbf{s}_t$ and the one-hot vector of the previous ground-truth symbol $\mathbf{y}_{t-1}$ using the following equation:

$$\mathbf{P}(\mathbf{y}_t | \mathbf{y}_{t-1}, \mathbf{X}) = g(\mathbf{W}_o h(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{W}_s\mathbf{s}_t + \mathbf{W}_c\mathbf{c}_t)) \tag{12}$$

where g denotes a softmax activation function over all the symbols in the vocabulary, and h denotes a maxout activation function. Let m' denote the dimension of embedding, $\mathbf{W}_o \in \mathbb{R}^{K \times \frac{m'}{2}}$, $\mathbf{W}_s \in \mathbb{R}^{m' \times n}$, $\mathbf{W}_c \in \mathbb{R}^{m' \times C}$, and $\mathbf{E}$ denotes the embedding matrix.

5. Network architecture for radical-based Chinese text recognition

As introduced previously, RAN can be easily extended from recognizing single characters to recognizing text lines. As illustrated in Fig. 4, by inserting an end-of-character (eoc) sentinel between every two Chinese character IDS sequences, RAN can predict the IDS sequence of the whole text line. We then transfer each IDS sequence, divided by eoc sentinels, into its Chinese character category using the IDS dictionary. To achieve better performance for robust text line recognition, we improve the encoder by employing a denseRNN encoder with HSV representations and exploit a multihead coverage-based attention mechanism.

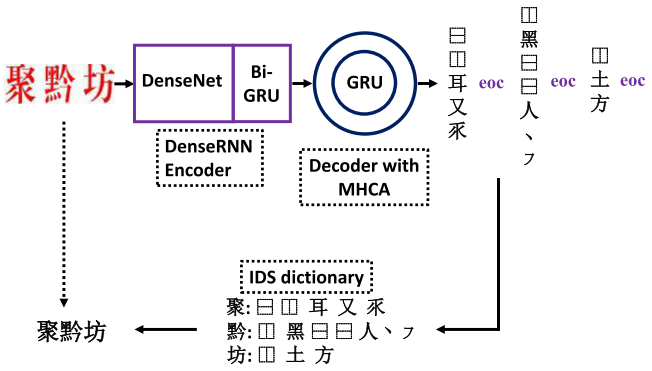


Fig. 4. Illustration of the extension of RAN for text line recognition. The method includes a denseRNN encoder and a GRU decoder equipped with multihead coverage attention (MHCA). An end-of-character (eoc) flag is added between every two adjacent IDS sequences for separation.

Fig. 5. Illustration of the benefits of HSV channels. Two text line images are difficult to recognize via RGB representations, but the representation can be improved by using HSV channels. The red "H" denotes that the image is the visualization of the H channel, while the red "S" denotes the S channel. The groundtruth, recognition results without HSV channels and recognition results with HSV channels are shown right behind "Label", "RGB" and "HSV".

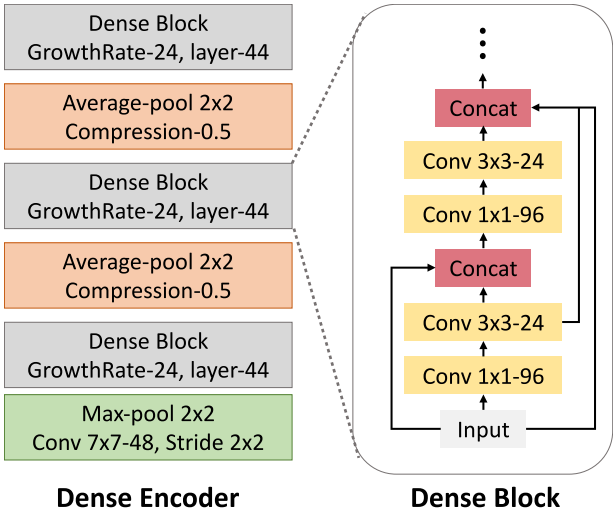


Fig. 6. Detailed architecture of the dense encoder, which we call the DenseNet135 encoder since it includes a total of 135 convolutional layers.

5.1. DenseRNN encoder with HSV representations

5.1.1. DenseRNN encoder

The features extracted from the CNN are directly used for character recognition, whereas a stack of RNNs, called the denseRNN architecture, is built on top of the convolutional layers to capture the context information in the text line for text line recognition. As shown in Eq. (2), a GRU is a parameterized RNN function that recursively maps an input vector and a hidden state to a new hidden state; hence, the GRU can capture the historical context. Accordingly, we pass the CNN features through GRU layers and use the output of the GRU layers as the new features A.

The extracted CNN features form a grid $\mathbf{V}$ of size $H \times W \times C$, where H denotes the number of rows, W denotes the number of columns and C denotes the number of feature maps. We first split grid $\mathbf{V}$ into H rows $\mathbf{V} = [\overline{\mathbf{V}}_1, \ldots, \overline{\mathbf{V}}_H]^{\mathrm{T}}$, where each $\overline{\mathbf{V}}_h$ is a sequence of length W, $\overline{\mathbf{V}}_h = [\overline{\mathbf{v}}_{h1}, \ldots, \overline{\mathbf{v}}_{hW}]$, $\overline{\mathbf{v}}_{hw} \in \mathbb{R}^C$. The new feature grid $\mathbf{A} = [\overline{\mathbf{A}}_1, \ldots, \overline{\mathbf{A}}_H]^{\mathrm{T}}$ is created from $\mathbf{V}$ by running the GRU function across each row. Recursively for all $\overline{\mathbf{A}}_h = [\overline{\mathbf{a}}_{h1}, \ldots, \overline{\mathbf{a}}_{hW}]$, $\overline{\mathbf{a}}_{hw} = \mathrm{GRU}\big(\overline{\mathbf{v}}_{hw}, \overline{\mathbf{a}}_{h(w-1)}\big)$. Nevertheless, a unidirectional GRU cannot utilize the future context. To implement a bidirectional GRU, we pass the input vectors through two GRU layers running in opposite directions and concatenate their hidden state vectors so that the new features $\mathbf{A}$ can capture both historical and future information. Features that can capture context information are crucial for the good performance of RAN in text line recognition, as some ambiguous characters are easier to distinguish when observing their contexts. The denseRNN can be jointly trained in a unified network and can operate on text line images of arbitrary size by traversing from start to end.

5.1.2. HSV representation

For Chinese character recognition in web images, the background is sometimes excessively noisy, and the input text line is difficult to recognize in RGB representations, even for human eyes, as shown in the examples in Fig. 5. Therefore, in addition to RGB representations, we use HSV representations to improve the visibility of text in web images. HSV representations include three channels, hue (H), saturation (S) and value (V), which are designed to more closely align with the way that humans perceive color-making attributes. As illustrated in Fig. 5, when RGB channels provide an ambiguous representation, HSV channels can give a much clearer visual image, leading to much better recognition results.

5.2. Multihead coverage-based attention (MHCA)

Multihead attention was first explored in [58] for machine translation, and we extend it to improve our RAN's performance on text line recognition tasks. Specifically, MHCA extends the conventional coverage-based attention mechanism to have multiple heads, where each head can generate a different attention distribution. This process enables each head to play a different role in attending to the encoder output, which we hypothesize makes it easier for the decoder to learn to retrieve information from the encoder. In the conventional, single-head architecture, the model relies on the encoder to provide clear signals about the sentences so that the decoder can obtain information via attention. We hypothesize that MHCA reduces the burden on the encoder and can distinguish radicals from noisy backgrounds when the encoded representation is less than ideal. The MHCA employs M independent attention heads, each of which computes a context vector $\mathbf{c}_t^m, 1 \le m \le M$:

$$\mathbf{F} = \mathbf{Q} * \sum_{l=1}^{t-1} \boldsymbol{\alpha}_l \tag{13}$$

$$e_{ti}^m = \boldsymbol{\nu}_{\mathrm{att}}^{m\,\mathrm{T}} \tanh(\mathbf{W}_{\mathrm{att}}^m \hat{\mathbf{s}}_t + \mathbf{U}_{\mathrm{att}}^m \mathbf{a}_i + \mathbf{U}_f^m \mathbf{f}_i) \tag{14}$$

$$\alpha_{ti}^m = \frac{\exp(e_{ti}^m)}{\sum_{k=1}^{H\times W} \exp(e_{tk}^m)} \quad \mathbf{c}_t^m = \sum_{i=1}^{H\times W} \alpha_{ti}^m \mathbf{Z}^m \mathbf{a}_i \tag{15}$$

Here, $\boldsymbol{\alpha}_l$ is the attention map of size $H \times W \times M$; the convolution filter Q has M input channels and q output channels. n and n' de-

note the dimensions of the GRU decoder and original single-head attention. Let $C^{'}$ denote the dimensions of the new feature vectors A; then, $\boldsymbol{v}_{\text{att}}^{m} \in \mathbb{R}^{\frac{n'}{M}}$, $\mathbf{W}_{\text{att}}^{m} \in \mathbb{R}^{\frac{n'}{M} \times n}$, $\mathbf{U}_{\text{att}}^{m} \in \mathbb{R}^{\frac{n'}{M} \times C^{'}}$, $\mathbf{U}_{f}^{m} \in \mathbb{R}^{\frac{n'}{M} \times q}$, $\mathbf{Z}^{\mathbf{m}}$ is the projection matrix of each head, $\mathbf{Z}^{\mathbf{m}} \in \mathbb{R}^{\frac{C^{'}}{M} \times C^{'}}$. The final context vector is computed by concatenating the individual summaries: $\mathbf{c}_t = [\mathbf{c}_t^1; \mathbf{c}_t^2; \cdots; \mathbf{c}_t^M]$. In our experiments, we propose 4 heads ($M = 4$).

6. Training and testing procedures

6.1. Training

During the training procedure, RAN aims to maximize each predicted symbol probability by utilizing cross-entropy as the objective function, $O = -\sum_{t=1}^{T} \log p(w_t|\mathbf{y}_{t-1}, \mathbf{X})$, where w_t represents the ground-truth symbol at time step t, $\mathbf{y}_{t-1}$ denotes the one-hot vector of the previous ground-truth symbol, $\mathbf{X}$ denotes the input character image, and T denotes the length of the output sequence.

The details of our dense encoder are presented in Fig. 6. The left part of Fig. 6 shows that we employ three dense blocks in the main branch. Before entering the first dense block, a 7×7 convolution (stride is 2×2) with 48 output channels is performed on the input expression images, followed by a 2×2 max pooling layer. We use 1×1 convolution followed by 2×2 average pooling as compression layers to reduce the feature maps by half. The right part of Fig. 6 shows the details of each dense block. Each dense block is labeled "DenseB" because we use bottleneck layers to improve the computational efficiency, i.e., a 1×1 convolution is introduced before each 3×3 convolution to reduce the input to $4k$ feature maps. The input of each bottleneck convolution is the concatenation of all previous 3×3 convolution output feature maps. The growth rate $k = 24$ and the depth (number of convolution layers) of each block $D = 44$, which means each block has 22 1×1 convolution layers and 22 3×3 convolution layers. A batch normalization layer and a ReLU activation layer are placed consecutively after each convolution layer. We call the encoder DenseNet135 since a total of 135 convolution layers are included in the framework. For the denseRNN encoder architecture, the CNN part of denseRNN has the same architecture as that of DenseNet135, the RNN part employs a stack of two bidirectional GRU layers, with each layer containing 256 forward GRU units and 256 backward GRU units ($C^{'} = 512$).

The decoder adopts 2 unidirectional GRU layers, and each layer has 256 forward GRU units. The embedding dimension m' and GRU decoder dimension n are set to 256. The attention dimension n' and the output channels of coverage convolution q for the single-head coverage attention model are set to 512. In the multihead coverage attention model, since we employ 4 heads, the attention dimension for each head is 128. We employ the adadelta algorithm [59] for optimization. The adadelta hyperparameters are set to $\rho = 0.95$ and $\varepsilon = 10^{-6}$.

6.2. Testing

During testing, RAN aims to generate the most likely IDS sequence given the input image:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \log P(\mathbf{y}|\mathbf{x}) \tag{16}$$

Unlike for the training procedure, we do not have the ground truth of the previous predicted symbol when predicting the IDS sequence. To alleviate the problem of mismatch between the training and predicting procedures, a simple left-to-right beam search algorithm [60] is employed to implement the prediction procedure. In each time step, we maintain a set of 5 partial hypotheses. Each partial hypothesis in the beam is expanded with every possible symbol, and only the 5 most likely beams are kept. The prediction procedure for each hypothesis ends when the output symbol reaches the end of the sequence. After successfully predicting the IDS sequence, we recognize the input Chinese character by searching the predefined IDS dictionary to find the character category whose IDS sequence is most like the predicted IDS sequence. We define the similarity between two IDS sequences using the minimum edit distance.

The adoption of an ensemble beam search procedure [61] is intuitive for improving performance. We first train N^e RAN models on the same training set with different initialized parameters. Then, we average their prediction probabilities to predict the current output symbol.

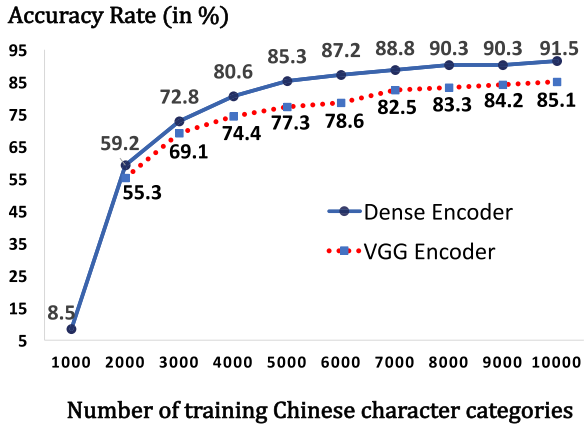


Fig. 7. Recognition performance of RAN for 17,533 unseen Chinese character categories with respect to the number of Chinese characters in the training samples.

7. Experiments on Chinese character recognition

7.1. Experiments on unseen character recognition

In this section, we first demonstrate the effectiveness of RAN for recognizing unseen Chinese character categories. The GB18030 standard includes 27,533 Chinese characters that are composed of only 485 radicals. We choose 10,000 character categories as the training set and use the other 17,533 character categories as the testing set. Clearly, the Chinese character categories in the testing set have not been seen during training. Both the training and testing inputs use Chinese character images in Song font style. The input images have the size of 32×32. Each Chinese character category only has one sample.

7.1.1. Accuracy versus character categories

Note that traditional character-based methods fail to recognize unseen Chinese characters, which means that their accuracies are **0%** in this experiment. However, RAN can recognize the unseen characters by predicting their IDS sequences and searching for the correct character categories in the IDS dictionary. We increase the training set from 1,000 to 10,000 Chinese characters to see how many Chinese characters are sufficient for training RAN to recognize the remaining unseen 17,533 characters. We illustrate the performance in Fig. 7. RAN (using DenseNet135 encoder) trained on 2,000 Chinese characters successfully recognizes 59.2% of the 17,533 unseen Chinese characters, and RAN trained on 10,000 Chinese characters achieves an accuracy of 91.5%, which means that 10,000 Chinese characters can help RAN recognize more than 16,000 of the unseen Chinese character categories. Approximately 500 Chinese characters can be used to sufficiently cover the 485

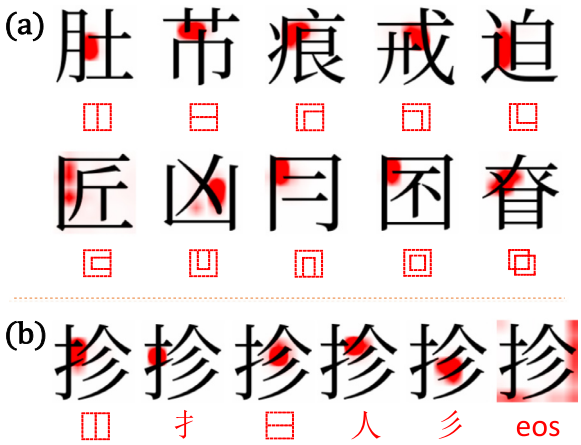


Fig. 8. Attention probabilities are shown in red: a darker red denotes a higher probability and a lighter red denotes a lower probability. (a) Attention visualization for recognizing 10 common radical structures; symbols below the images are the predicted radical structures. (b) Attention visualization for recognizing a Chinese character instance; symbols below the images are the predicted radicals or structures.

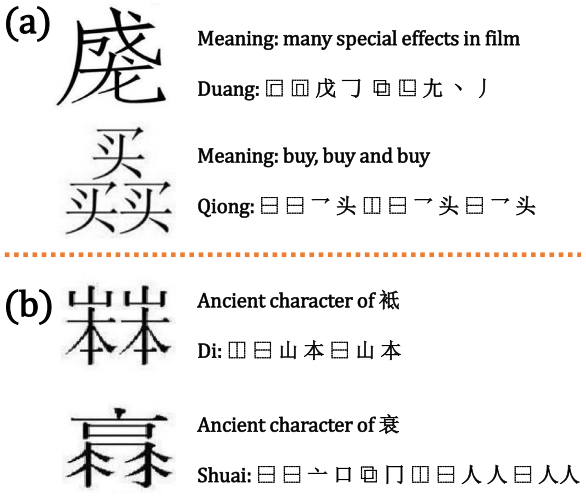


Fig. 9. (a) Recognition of newly created Chinese characters from the internet; (b) Recognition of rarely used ancient Chinese characters.

Chinese radicals and 10 spatial structures. However, our experiments start with 1,000 training characters because RAN has difficulty converging when the training set is too small.

By comparing the results of RAN using the proposed DenseNet135 encoder and RAN using the proposed VGG encoder in [20], we clearly see better visual features extracted from Chinese character images can help improve the zero-shot learning ability of RAN.

7.1.2. Attention visualization

Unlike conventional radical-based Chinese character recognition methods, RAN employs an attention model to segment radicals and identify the structures among the segmented radicals. Here, we prove that the attention model can achieve human-like radical alignment and structure detection through attention visualization. In Fig. 8(a), we present 10 examples of how RAN identifies structures for every pair of radicals. The red color in the attention maps represents the spatial attention probability, where a darker red indicates a higher attention probability and a lighter red indicates a lower attention probability. Taking the left-to-right structure as an example, the attention model focuses on the space between the two horizontal radicals, which implicitly indicates a left-right direction. When identifying the above-to-below structure, the attention model focuses on the space between the two vertical radicals, which implicitly indicates an above-to-below direction. The focus of attention corresponds well to human intuition when identifying other radical structures. More specifically, in Fig. 8(b), we show the step-by-step process of RAN learning to recognize an unseen Chinese character in an IDS sequence. When encountering basic radicals, the attention model generates an alignment that strongly corresponds to human intuition and successfully predicts the radical structures "LR" and "AB" when a left-to-right direction and an above-to-below direction are detected, respectively.

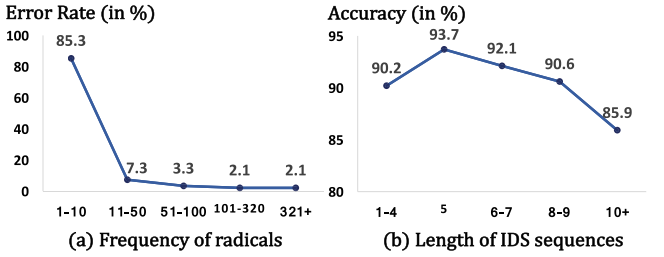


Fig. 10. (a) Radical-level error rate; each point describes the rate of all radicals within a specific range of appearance frequency that are substituted or deleted in the output IDS sequences. The range on the horizontal axis denotes the number of times that these radicals appear in the Chinese character training set. Approximately 100 radical categories are included in each range. (b) Character-level accuracy with respect to the length of the IDS sequences. Approximately 3,000 character categories are included in each range.

7.1.3. Examples of zero-shot learning

Fig. 9 illustrates how RAN can be used to recognize newly created Chinese characters and rarely used Chinese characters through zero-shot learning. For example, Fig. 9(a) shows two novel Internet Chinese characters: "Duang", which was created by Jackie Chan, meaning many special effects in film, and "Qiong", meaning being poor due to too many expenses. The generated hierarchical radical structures correspond well to human intuition. Fig. 9(b) shows two rarely used ancient characters. Although these Chinese characters have not been previously observed, RAN can successfully recognize them by adding the new correspondence between IDS sequences and related characters into the new IDS dictionary.

7.1.4. Error analysis

Note that nearly 1,490 Chinese characters are still misrecognized when RAN is trained on 10,000 Chinese characters. In Fig. 10, we analyze the cause of incorrect recognition. We first analyze the frequency of the radicals that fail to be recognized (including substitution errors and deletion errors). As introduced in Section 3, although RAN helps alleviate the problem of recognizing low-frequency Chinese characters, some low-frequency radicals still cause difficulties for few-shot learning. Fig. 10(a) shows that radicals that appear fewer than 10 times are highly likely to be incorrectly recognized due to lack of learning. By contrast, for high-frequency radicals, the error rate is approximately 2% because they are shared by different Chinese characters in the training samples and have been learned sufficiently during training. Another interesting result is the distribution of accuracy with respect to the length of the IDS sequences. We expect the model to perform poorly on Chinese characters with longer IDS sequences since the characters that need to be decomposed into longer IDS sequences are usually related to more complicated structures and composed of more radicals, hence increasing the difficulty of structure detection and radical alignment. Fig. 10(b) illustrates this behavior.

Table 1
Comparison of the performance of powerful image classifiers and RAN on the CTW test database. We divide the Chinese character categories into 4 subsets based on the appearance frequency in the training set. OOV represents out-of-vocabulary, i.e., character categories that are not included in the training set; <20 indicates character categories that appear fewer than 20 times; <100 indicates character categories that appear fewer than 100 times; HF stands for high frequency and includes character categories that appear more than 100 times; and ALL includes all character categories in the testing set.

Frequency	OOV	<20	<100	HF	ALL
Categories	70	328	573	1044	2015
Samples	182	946	2892	48745	52765
AlexNet	0%	24.4%	47.3%	77.1%	74.3%
ResNet50	0%	24.4%	53.5%	81.3%	78.5%
ResNet152	0%	22.2%	55.5%	81.6%	78.8%
DenseNet135	0%	25.3%	55.8%	82.9%	80.1%
RAN	**19.6%**	**35.2%**	**59.2%**	**84.3%**	**81.8%**

7.2. Experiments on low-frequency character recognition

In this section, we do experiments to illustrate the effectiveness of RAN for recognizing low-frequency Chinese characters and compare RAN with powerful image classifiers. To explore the practical value of RAN, we investigate the performance on a task on the recognition of Chinese character in the wild.

7.2.1. CTW database

Our experiments are performed on the CTW dataset [62], which contains Chinese character images collected from street views. The dataset is challenging due to its diversity and complexity. It contains planar text, raised text, text in cities, text in rural areas, text under poor illumination, distant text, partially occluded text, etc. Moreover, many low-frequency Chinese character categories are included. Because the official testing set is not released, we use the official validation set as our testing set for analysis.

The CTW database contains 3,580 Chinese character categories with 760,107 instances for training and 2,015 Chinese character categories with 52,765 instances for testing. All input images have the size of 32×32. Table 1 presents a detailed comparison of RAN and other character-based methods on the CTW database. We divide all testing character categories into 4 subsets based on the appearance frequency in the CTW training set to clearly demonstrate the effectiveness of RAN for few-shot learning. A total of 70 character categories are not observed in the training set, and 328 character categories have fewer than 20 training samples. Therefore, the recognition of these Chinese characters is a few-/zero-shot learning problem, which is difficult due to the limited number of training samples.

7.2.2. Comparison to character-based classifiers

We tested several state-of-the-art character-based classifiers, namely, AlexNet [63], ResNet50 [64], and ResNet152, using Pytorch. To ensure a fair comparison, we also train a DenseNet classifier, named DenseNet135, with the same CNN architecture as that of the dense encoder of RAN. Compared with the 0% OOV recognition of DenseNet135, the 19.6% recognition rate of RAN proves that RAN maintains the zero-shot learning ability in natural scenes. Moreover, RAN improves low-frequency character recognition (<20) by nearly 10% and improves the recognition of characters that appear fewer than 100 times in the training set (<100) by nearly 4%. The recognition of RAN for unseen Chinese characters (OOV) and low-frequency Chinese characters (<20) in natural scenes is not as good as that for printed Chinese characters. We show several failed instances of zero-shot learning in Fig. 11(a); these instances are incorrectly recognized because they are challenging even for humans.

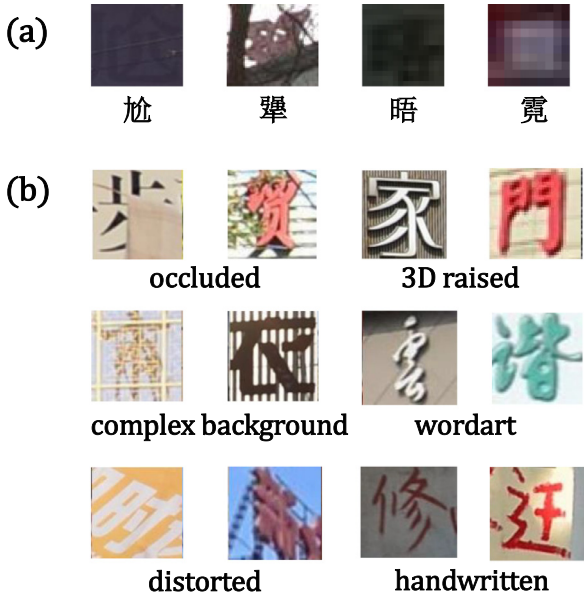


Fig. 11. (a) Failed examples of zero-shot learning for RAN in the CTW testing set; (b) Visualization of 6 difficult attributes in the CTW database; each attribute shows 2 examples.

Table 2
Comparison of the recognition performance of the DenseNet image classifier and RAN on the CTW test database with respect to 7 attributes: clean, occluded, complex background, distorted, 3D raised, wordart characters and handwritten characters.

Attributes	Training samples	DenseNet135	RAN
Clean	273831	86.5%	87.2%
Occluded	101393	67.6%	70.7%
Background	218560	76.4%	78.7%
Distorted	192481	76.3%	77.7%
3D raised	199066	77.6%	80.6%
Wordart	65983	68.9%	72.7%
Handwritten	6661	59.4%	68.2%

RAN can also improve the recognition rate of high-frequency Chinese characters by approximately 1.4%. We believe that the improvement is due to the following: (i) RAN decreases the size of the output dictionary, reduces the redundancy among similar Chinese characters and makes the model easier to train properly; and (ii) RAN increases the robustness of the recognition model to noisy and complex images in natural scenes because radicals can be shared by many Chinese characters. More variation and transformation information can be learned if the training objective instances are radicals rather than characters.

7.2.3. Analysis of robustness

To demonstrate the robustness of RAN in natural scenes, Table 2 compares the performance of DenseNet135 and RAN with respect to 7 attributes, where clean means that there is no noise in the image and that the character in the image has no transformation. We visualize the occluded, complex background, distorted, 3D raised, wordart and handwritten attributes in Fig. 11(b). The handwritten Chinese characters are the most difficult to recognize because they are more likely to ignore the internal structures of Chinese characters. The recognition results of DenseNet135 and RAN shows that RAN achieves only a 0.7% improvement on the clean attribute since the clean character images are not challenging and the number of low-frequency Chinese characters is small. However, RAN achieves considerable improvement compared with DenseNet135 on the dif-

Table 3
Detailed analysis of the Chinese character distribution in the ICPR MTWI 2018 database. We divide the testing set into 4 subsets based on the frequency of appearance of the Chinese characters in the text lines in the training set. We report the number of text lines, the number of Chinese character instances and the number of Chinese character categories in each subset.

Set		Line samples	Char samples	Char categories
Train		76130	298550	4010
Test	OOV	120	134	76
	< 20	921	1180	597
	< 100	2346	3469	732
	HF	5801	19256	683
	ALL	6129	24039	2088

ficult attributes, especially wordart and handwritten. We believe that the improvement is because the wordart and handwritten training samples are insufficient for training a character-based classifier, whereas for RAN, a radical can be shared by several Chinese characters. Hence, radicals are several times more frequent than characters in the training instances of wordart and handwritten, leading to improvements of 3.8% on wordart and 8.8% on handwritten.

Among these attributes, handwritten is a special one. More specifically, in [22,23], we have a detailed discussion about how well RAN performs on handwritten Chinese character recognition. Since different people have different writing styles and writing habits, handwriting input brings much ambiguities and usually loses radicals. In experiments, RAN performs bad on recognizing unseen handwritten Chinese characters because it cannot detect radicals when radicals are missing in handwriting input. While for recognizing seen handwritten Chinese characters, the missing radicals will not have much influence as RAN can rely on language model to predict the missing radical.

8. Experiments on Chinese text line recognition

8.1. Text dataset

8.1.1. ICPR MTWI-18

The ICPR MTWI-18 text line recognition challenge is one of the largest published databases for Chinese text line recognition. The database of the ICPR MTWI challenge is collected from web images and includes various font styles and complex backgrounds. Although the database includes English characters, Chinese characters are dominant. The official training set contains 128,210 text lines, with Chinese characters included in 76,130 text lines. A total of 4,010 Chinese character categories with 298,550 character instances are contained in the training set. As the official testing set is not released, we use the official validation set as our testing set, which includes 2,088 Chinese character categories with 24,039 character instances. Similar to the analysis of the CTW database, we present a detailed illustration of OOV and low-frequency Chinese characters in the MTWI database in Table 3. We resize all text line images by turning the shorter length of images into 32 and keeping the ration between height and width unchanged.

As shown in Table 3, 76 OOV Chinese character categories with 134 Chinese character instances are included in 120 text lines. Additionally, 597 Chinese character categories appear fewer than 20 times in the training set, with 1,180 character instances in 921 text lines. Therefore, the most challenging part of this task is that it requires a system with zero-/few-shot learning ability since nearly 11% of the testing text lines contain low-frequency Chinese characters. Most participants in this competition utilize character-based text line recognition methods, and their deep learning models fail to recognize these low-frequency Chinese characters if they use only the official training database. As a result, RAN's zero-/few-shot learning ability showed great power on this challenge and helped us win first place in the ICPR MTWI 2018 challenge.

Table 4
Detailed comparison of the text line recognition performance of CTC, character-based encoder-decoder and RAN. We divide the MTWI testing set into 4 subsets: out-of-vocabulary (OOV), < 20, < 100 and high frequency (HF). CER is the character error rate; SACC is the whole sentence accuracy.

System		CRNN	SCCM	Encoder-Decoder	RAN
CER	OOV	100%	100%	100%	80.6%
	< 20	69.2%	67.6%	65.5%	44.9%
	< 100	32.7%	33.5%	31.4%	21.3%
	HF	12.2%	12.9%	10.2%	9.5%
	ALL	18.5%	19.0%	16.5%	13.3%
SACC	OOV	0%	0%	0%	5%
	< 20	24.7%	25.1%	26.9%	39.4%
	< 100	41.9%	40.2%	45.8%	57.1%
	HF	64.2%	62.9%	66.9%	68.2%
	ALL	58.8%	57.9%	61.8%	67.3%

8.1.2. RCTW-17

RCTW-17 is a challenge that involves reading Chinese text found in the real world [18]. There are 44,009 text line images in the training set. The images are collected from street views, posters, menus, indoor scenes and screenshots. Because the testing set is not released by the official organization team and there is no official validation set, we cannot analyze the frequency of Chinese character categories. However, an overall recognition performance is obtained to help us further prove the superiority of the proposed RAN. In addition, RCTW-17 is an end-to-end text line recognition problem, which means that we need to detect text lines from large images before recognizing text lines.

8.2. Evaluation of RAN on ICPR MTWI-18

8.2.1. Metric

In Table 4, we compare RAN with other character-based text line recognition methods on CER and SACC. Here, CER and SACC count only the performance of Chinese characters, regardless of English or other characters.

CER is the character error rate. As we count the errors of specific characters, there are no insertion errors, and only substitution and deletion errors are included. For example, CER-OOV denotes the results of the character error rate for recognizing OOV Chinese characters.

SACC is the sentence accuracy rate. We count only correct sentences containing specific characters. For example, SACC-OOV denotes the accuracy rate of whole text lines containing OOV Chinese characters.

8.2.2. Systems

We compare RAN with the **SCCM-CTC** (sliding convolution character model), **CRNN-CTC** model and **encoder-decoder** (character-based encoder-decoder model). The SCCM model exploits the architecture of [43] and the CRNN model exploits the architecture of [42], but we improve the model by increasing the number of feature maps. We implement another 3-gram language model trained on the official text database for SCCM and CRNN. The encoder-decoder model exploits the same architecture as RAN but is modeled on the character level. Therefore, comparison of RAN with encoder-decoder provides an improved understanding of the advantage of RAN, as these methods are comparable.

8.2.3. Performance

The character-based approaches fail to recognize text lines containing OOV Chinese characters, leading to **0%** SACC and **100%** CER.

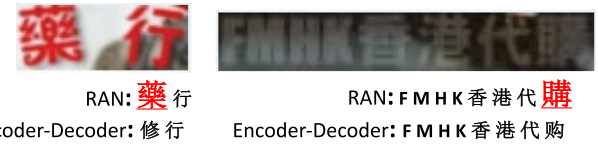


Fig. 12. Two examples of text lines containing low-frequency Chinese characters (underlined and shown in red). The recognition results predicted by the proposed RAN and the character-based encoder-decoder are shown below the text line images.

By contrast, RAN can recognize unseen Chinese characters if the radicals have already been seen, resulting in an improvement of 19.4% on CER and an improvement of 5% on SACC. By comparing the CER of the encoder-decoder and RAN on high-frequency Chinese characters, we can see that the proposed RAN still achieves remarkable improvement since RAN is more robust to complex backgrounds and character variation than the character-based approaches. RAN's advantage of recognizing low-frequency Chinese characters is clearly observed by comparing the results of RAN and the character-based encoder-decoder model on the $<$ 20 and $<$ 100 subsets.

8.2.4. Efficiency

It is necessary to compare the recognition speed between RAN and Encoder-Decoder as we can see whether RAN will largely increase the decoding time since it enlarges the length of decoding steps. The efficiency of RAN and Encoder-Decoder are totally comparable because they have the same architecture and are implemented using the same tool. RAN and Encoder-Decoder are implemented with Theano 0.10.0 and an NVIDIA Tesla M40 24G GPU, while CRNN and SCCM are implemented with Pytorch. Here, we introduce the average time cost of RAN and Encoder-Decoder for recognizing each character on all 15,288 text lines with a testing batch size of 1. Using the same denseRNN encoder, RAN needs 11.33 ms on average and Encoder-Decoder needs 11.37ms on average, we can see RAN is even slightly faster than Encoder-Decoder. It is because although RAN enlarges the length of decoding steps, at each step, RAN needs less computation cost when performing softmax operation than Encoder-Decoder since the number of output classes reduces from 4,010 to 495.

8.2.5. Examples of few-shot learning

Fig. 12 shows two examples of text lines containing low-frequency Chinese characters. The two characters in red are low-frequency characters that appear only a few times in the training database. These characters belong to complex traditional Chinese character categories that are rarely used in common scenes and therefore unlikely to be in the training set. The character-based encoder-decoder approach fails to recognize these characters but RAN successfully recognizes them, as they are composed of basic structures whose essential radicals have already been learned.

8.3. Evaluation of RAN on RCTW-17

Table 5 shows the experimental results on the RCTW-17 database. System NLPR-PAL, SCUT-DLVC, and CCFLAB are the top-3 systems in the challenge. Since their text detectors are different, their text recognition systems are not totally comparable. However, we can see that using an extra synthetic dataset has a great effect on the recognition performance because the training set is small and there are many unseen and low-frequency Chinese characters in the testing set.

For comparability, system SCCM, CRNN, encoder-decoder and RAN employed the same text detector, i.e., TextMountain [65], whose detection performance is similar to the one used in the NLPR-PAL system. Because we did not use synthetic data, the cognition performance of RAN is worse than that of NLPR-PAL. However, we can still see that RAN is the best method when using only official training data, indicating its superiority on zero/few-shot learning tasks.

Table 5
Comparison of average edit distance (AED) on the RCTW-17 testing set. Synthetic Data denotes whether the system used extra synthetic data for training.

System(*)	Synthetic data	AED
NLPR-PAL	✓	20.2
SCUT-DLVC	✓	28.3
CCFLAB	×	32.1
Baseline	✓	25.6
SCCM	×	26.5
CRNN	×	25.3
Encoder-Decoder	×	24.8
RAN	×	22.8

Table 6
Comparison of CER and SACC for both Chinese and other characters/texts (in %) and the time efficiency (in ms) on the MTWI testing set when appending the denseRNN encoder, MHCA and HSV channels to the proposed RAN system. * indicates an ensemble model.

System(*)	CER	SACC	Time
RAN	14.8%	63.1%	10.4 ms
+ denseRNN	12.5%	66.6%	11.3 ms
+ MHCA	11.7%	68.1%	11.6 ms
+ HSV	11.1%	68.6%	11.6 ms

8.4. Evaluation of the proposed denseRNN, MHCA and HSV

8.4.1. Performance

Table 6 shows the improvements via the denseRNN encoder, MHCA and HSV representations by appending each to the previous system. We present the results of the ensemble models, and the number of combined models N^e is set to 5.

First, the system "+ denseRNN" adds a new bidirectional GRU encoder immediately after the DenseNet encoder not only to extract high-level visual features from the input images but also to capture the context information in the text lines. The input image of the denseRNN encoder can be an arbitrary size. The denseRNN encoder decreases CER from 14.8% to 12.5% and improves SACC from 63.1% to 66.6%.

Second, CER is further decreased from 12.5% to 11.7% after the single-head coverage attention is replaced by MHCA, and SACC is increased by 1.5%, which indicates that the attention model with multiple heads generates a better attention distribution than that with a single head.

Finally, consideration of the HSV information of the color images embedded in the input channels decreases CER from 11.7% to 11.1%, which plays an important role in strengthening the ability of RAN for distinguishing Chinese characters against very complex backgrounds. Also, comparing the recognition results of RAN with and without HSV information in Fig. 5, it is clear to see that the recognition results are improved by using HSV channels.

8.4.2. Efficiency

We also compare the computational costs of the above systems by investigating their speeds. We present the average time cost

for recognizing each character on all 15,288 text lines with a testing batch size of 1. Appending the new bidirectional GRU encoder after the DenseNet encoder slows the average test speed for one text line by 5 ms despite the considerable improvement in recognition performance. The MHCA and HSV representations have a minimal effect on the test speed because the total number of parameters in the attention model does not change when switching from a single head to multiple heads, and the computational cost of adding 3 input channels to the first convolutional layer can be ignored.

9. Conclusion and future work

In this study, we introduce a novel radical analysis network for radical-based Chinese character and Chinese text line recognition. The proposed model imitates the technique used by Chinese learners to recognize Chinese characters. We demonstrate through visualization and experimental results that RAN has the ability to use few-/zero-shot learning to learn Chinese characters. Additionally, we present detailed comparisons to demonstrate RAN's advantages in the recognition of low-frequency character categories in both single-character recognition and text line recognition. We also verify the practical value of RAN in natural scenes. The released IDS dictionary will benefit related studies.

In future work, we plan to identify a better method for the decomposition of Chinese characters, and we will improve the attention model to increase the few-/zero-shot learning ability of RAN for recognizing low-quality Chinese character images. We hope that by proposing a novel radical-based recognition model, people will be encouraged to create more interesting and personal Chinese characters, as novel characters can be easily recognized.

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.

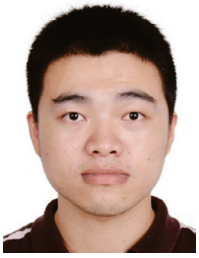
Acknowledgment

This work was supported in part by the National Key R and D Program of China under contract No. 2017YFB1002202, the National Natural Science Foundation of China under Grant Nos. 61671422 and U1613211, the Key Science and Technology Project of Anhui Province under Grant No. 17030901005. This work was also funded by Huawei Noah's Ark Lab.

References

[1] Y.Y. Tang, L.-T. Tu, J. Liu, S.-W. Lee, W.-W. Lin, Off-line recognition of Chinese handwriting by multifeature and multilevel classification, IEEE Trans. Patt. Anal. Mach. Intell. 20 (5) (1998) 556–561.
[2] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.
[3] J.D. Allen, D. Anderson, J. Becker, R. Cook, M. Davis, P. Edberg, M. Everson, A. Freytag, L. Iancu, R. Ishida, et al., The Unicode Standard, 6, Citeseer, 2012.
[4] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[5] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[6] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1724–1734.
[7] C.-L. Liu, S. Jaeger, M. Nakagawa, Online recognition of Chinese characters: the state-of-the-art, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2) (2004) 198–213.
[8] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, Online and offline handwritten Chinese character recognition: benchmarking on new databases, Pattern Recognit. 46 (1) (2013) 155–162.
[9] D. George, W. Lehrach, K. Kansky, M. Lázaro-Gredilla, C. Laan, B. Marthi, X. Lou, Z. Meng, Y. Liu, H. Wang, et al., A generative vision model that trains with high data efficiency and breaks text-based captchas, Science 358 (6368) (2017) eaag2612.
[10] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, IEEE Trans. Pattern Anal. Mach. Intell. 28 (4) (2006) 594–611.
[11] B.M. Lake, R. Salakhutdinov, J.B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (6266) (2015) 1332–1338.
[12] G. Sampson, Writing systems, London, 1985.
[13] Y. Suen, E. Huang, Computational analysis of the structural compositions of frequently used Chinese characters, Comput. Process. Chin. Oriental Lang. 1 (3) (1984) 163–176.
[14] M. Corbetta, G.L. Shulman, Control of goal-directed and stimulus-driven attention in the brain, Nature reviews neuroscience 3 (3) (2002) 201.
[15] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.
[16] K. Van De Sande, T. Gevers, C. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1582–1596.
[17] M. He, Y. Liu, Z. Yang, S. Zhang, C. Luo, F. Gao, Q. Zheng, Y. Wang, X. Zhang, L. Jin, ICPR2018 contest on robust reading for multi-type web images, in: International Conference on Pattern Recognition, 2018, pp. 7–12.
[18] B. Shi, C. Yao, M. Liao, M. Yang, P. Xu, L. Cui, S. Belongie, S. Lu, X. Bai, ICDAR2017 competition on reading Chinese text in the wild (RCTW-17), in: International Conference on Document Analysis and Recognition, 2017, pp. 1429–1434.
[19] H. Ni, GB18030 - the new Chinese encoding standard, from http://www.gb18030.com.
[20] J. Zhang, Y. Zhu, J. Du, L. Dai, Radical analysis network for zero-shot learning in printed Chinese character recognition, in: IEEE International Conference on Multimedia and Expo, 2018, pp. 1–6.
[21] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[22] J. Zhang, Y. Zhu, J. Du, L. Dai, Trajectory-based radical analysis network for online handwritten Chinese character recognition, in: International Conference on Pattern Recognition, 2018, pp. 3681–3686.
[23] W. Wang, J. Zhang, J. Du, Z.-R. Wang, Y. Zhu, DenseRAN for offline handwritten Chinese character recognition, in: International Conference on Frontiers in Handwriting Recognition, 2018, pp. 104–109.
[24] D. Cireşan, U. Meier, Multi-column deep neural networks for offline handwritten Chinese character classification, in: International Joint Conference on Neural Networks, 2015, pp. 1–6.
[25] C. Wu, W. Fan, Y. He, J. Sun, S. Naoi, Handwritten character recognition by alternately trained relaxation convolutional neural network, in: International Conference on Frontiers in Handwriting Recognition, 2014, pp. 291–296.
[26] Z. Zhong, L. Jin, Z. Xie, High performance offline handwritten Chinese character recognition using googlenet and directional feature maps, in: International Conference on Document Analysis and Recognition, 2015, pp. 846–850.
[27] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, Y. Bengio, Drawing and recognizing Chinese characters with recurrent neural network, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2018) 849–862.
[28] Y. Bengio, Y. LeCun, D. Henderson, Globally trained handwritten word recognizer using spatial representation, convolutional neural networks and hidden Markov models, in: Advances in Neural Information Processing Systems, 1994, pp. 937–944.
[29] B. Graham, Sparse arrays of signatures for online character recognition, 2013 arXiv:/1308.0371.
[30] W. Yang, L. Jin, D. Tao, Z. Xie, Z. Feng, Dropsample: A new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten Chinese character recognition, Pattern Recognit. 58 (2016) 190–203.
[31] D. Shi, S.R. Gunn, R.I. Damper, Handwritten Chinese radical recognition using nonlinear active shape models, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2) (2003) 277–280.
[32] F.-H. GHENG, W.-H. Hsu, Radical extraction from handwritten Chinese characters by background thinning method, IEICE Trans.(1976-1990) 71 (1) (1988) 88–98.
[33] S.W. Lu, Y. Ren, C.Y. Suen, Hierarchical attributed graph representation and recognition of handwritten Chinese characters, Pattern Recognit. 24 (7) (1991) 617–632.
[34] M. Zhao, Two-dimensional extended attribute grammar method for the recognition of hand-printed Chinese characters, Pattern Recognit. 23 (7) (1990) 685–695.
[35] D. Shi, R.I. Damper, S.R. Gunn, Offline handwritten Chinese character recognition by radical decomposition, ACM Trans. Asian Lang. Inf. Process. 2 (1) (2003) 27–48.
[36] A.-B. Wang, K.-C. Fan, Optical recognition of handwritten Chinese characters by hierarchical radical matching method, Pattern Recognit. 34 (1) (2001) 15–35.
[37] S.-R. Lay, C.-H. Lee, N.-J. Cheng, C.-C. Tseng, B.-S. Jeng, P.-Y. Ting, Q.-Z. Wu, M.-L. Day, On-line Chinese character recognition with effective candidate radical and candidate character selections, Pattern Recognit. 29 (10) (1996) 1647–1659.

[38] T.-Q. Wang, F. Yin, C.-L. Liu, Radical-based Chinese character recognition via multi-labeled learning of deep residual networks, in: International Conference on Document Analysis and Recognition, 2017, pp. 579–584.
[39] L.-L. Ma, C.-L. Liu, A new radical-based approach to online handwritten Chinese character recognition, in: International Conference on Pattern Recognition, 2008, pp. 1–4.
[40] Z.-R. Wang, J. Du, W.-C. Wang, J.-F. Zhai, J.-S. Hu, A comprehensive study of hybrid neural network hidden markov model for offline handwritten Chinese text recognition, Int. J. Doc. Anal. Recognit. 21 (4) (2018) 241–251.
[41] Z. Xie, Z. Sun, L. Jin, H. Ni, T. Lyons, Learning spatial-semantic context with fully convolutional recurrent network for online handwritten chinese text recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (8) (2018) 1903–1917.
[42] B. Shi, X. Bai, C. Yao, An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition, IEEE Trans. Pattern Anal. Mach. Intell. 39 (11) (2017) 2298–2304.
[43] F. Yin, Y.-C. Wu, X.-Y. Zhang, C.-L. Liu, Scene text recognition with sliding convolutional character models, 2017 arXiv:/1709.01727.
[44] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, X. Bai, Aster: An attentional scene text recognizer with flexible rectification, IEEE Trans. Pattern Anal. Mach. Intell. 41 (9) (2019) 2035–2048.
[45] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, S. Zhou, Focusing attention: Towards accurate text recognition in natural images, in: International Conference on Computer Vision, 2017, pp. 5076–5084.
[46] Z. Xie, Y. Huang, Y. Zhu, L. Jin, Y. Liu, L. Xie, Aggregation cross-entropy for sequence recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6538–6547.
[47] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3156–3164.
[48] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014 arXiv:/1409.0473.
[49] W. Chan, N. Jaitly, Q. Le, O. Vinyals, Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, in: International Conference on Acoustics, Speech and Signal Processing, 2016, pp. 4960–4964.
[50] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio, End-to-end attention-based large vocabulary speech recognition, in: International Conference on Acoustics, Speech and Signal Processing, 2016, pp. 4945–4949.
[51] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: International Conference on Machine Learning, 2015, pp. 2048–2057.
[52] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: IEEE conference on computer vision and pattern recognition, 2015, pp. 3128–3137.
[53] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, L. Dai, Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition, Pattern Recognit. 71 (2017) 196–206.
[54] J. Zhang, J. Du, L. Dai, Track, attend and parse (tap): An end-to-end framework for online handwritten mathematical expression recognition, IEEE Trans. Multimedia 21 (1) (2019) 221–233.
[55] J. Zhang, J. Du, L. Dai, Multi-scale attention with dense encoder for handwritten mathematical expression recognition, in: International Conference on Pattern Recognition, 2018, pp. 2245–2250.
[56] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Networks 5 (2) (1994) 157–166.
[57] Z. Tu, Z. Lu, Y. Liu, X. Liu, H. Li, Modeling coverage for neural machine translation, 2016 arXiv:/1601.04811.
[58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 6000–6010.
[59] M.D. Zeiler, Adadelta: an adaptive learning rate method, 2012 arXiv:/1212.5701.
[60] K. Cho, Natural language understanding with distributed representation, 2015 arXiv:/1511.07916.
[61] T.G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems, 2000, pp. 1–15.
[62] T.-L. Yuan, Z. Zhu, K. Xu, C.-J. Li, S.-M. Hu, Chinese text in the wild, 2018 arXiv:/1803.00085.
[63] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[64] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[65] Y. Zhu, J. Du, Textmountain: Accurate scene text detection via instance segmentation, 2018 arXiv:/1811.12786.



Jianshu Zhang received his B.Eng. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC) in 2015. He is currently a Ph.D. candidate of USTC. In 2018, he worked as a visiting student for 6 months in the Queen Mary University of London. His current research areas include deep learning, handwriting mathematical expression recognition, Chinese document analysis and speech analysis.

Jun Du received his B.Eng. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC) in 2004 and 2009, respectively. From 2004 to 2009, he was with iFlytek Speech Lab of USTC. During the above years, he worked as an Intern for two 9-month periods at Microsoft Research Asia (MSRA), Beijing. In 2007, he worked as a Research Assistant for 6 months in the Department of Computer Science, University of Hong Kong. From July 2009 to June 2010, he worked at iFLYTEK Research on speech recognition. From July 2010 to January 2013, he joined MSRA as an Associate Researcher, working on handwriting recognition, OCR, and speech recognition. Since February 2013, he has been with the National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP) of USTC.

Lirong Dai was born in China in 1962. He received a B.S. degree in electrical engineering from Xidian University, Xian, China, in 1983, his M.S. degree from the Hefei University of Technology, Hefei, China, in 1986, and a Ph.D. degree in signal and information processing from the University of Science and Technology of China (USTC), Hefei, in 1997. He joined USTC in 1993. He is currently a Professor at the School of Information Science and Technology, USTC. His research interests include speech synthesis, speech recognition, machine learning and pattern recognition.