2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Trajectory-based Radical Analysis Network for Online Handwritten Chinese Character Recognition

Jianshu Zhang, Yixing Zhu, Jun Du and Lirong Dai
National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, Anhui, P. R. China
Email: xysszjs@mail.ustc.edu.cn, zyxsa@mail.ustc.edu.cn, jundu@ustc.edu.cn, lrdai@ustc.edu.cn

***Abstract*—Recently, great progress has been made for online handwritten Chinese character recognition due to the emergence of deep learning techniques. However, previous research mostly treated each Chinese character as one class without explicitly considering its inherent structure, namely the radical components with complicated geometry. In this study, we propose a novel trajectory-based radical analysis network (TRAN) to firstly identify radicals and analyze two-dimensional structures among radicals simultaneously, then recognize Chinese characters by generating captions of them based on the analysis of their internal radicals. The proposed TRAN employs recurrent neural networks (RNNs) as both an encoder and a decoder. The RNN encoder makes full use of online information by directly transforming handwriting trajectory into high-level features. The RNN decoder aims at generating the caption by detecting radicals and spatial structures through an attention model. The manner of treating a Chinese character as a two-dimensional composition of radicals can reduce the size of vocabulary and enable TRAN to possess the capability of recognizing unseen Chinese character classes, only if the corresponding radicals have been seen. Evaluated on CASIA-OLHWDB database, the proposed approach significantly outperforms the state-of-the-art whole-character modeling approach with a relative character error rate (CER) reduction of 10%. Meanwhile, for the case of recognition of 500 unseen Chinese characters, TRAN can achieve a character accuracy of about 60% while the traditional whole-character method has no capability to handle them.**

I. Introduction

Machine recognition of handwritten Chinese characters has been studied for decades [1]. It is a challenging problem due to a large number of character classes and enormous ambiguities coming from handwriting input. Although some conventional approaches have obtained great achievements [2]–[6], they only treated the character sample as a whole without considering the similarity and internal structures among different characters. And they have no capability of dealing with unseen character classes.

However, Chinese characters can be decomposed into a few fundamental structure components, called radicals [7]. It is an intuitive way to first extract information of radicals that is embedded in Chinese characters and then use this knowledge for recognition. In the past few decades, lots of efforts have been made for radical-based Chinese character recognition. For example, [8] proposed a matching method for radical-based Chinese character recognition. It first detected radicals separately and then employed a hierarchical radical matching method to compose radicals into a character. [9] tried to over-segment characters into candidate radicals while the proposed way could only handle the left-right structure and over-segmentation brings many difficulties. Recently, [10] proposed a multi-label learning for radical-based Chinese character recognition. It turned a character class into a combination of several radicals and spatial structures. Generally, these approaches have difficulties when dealing with radical segmentation and the analysis of structures among radicals is not flexible. Besides, they did not focus on recognizing unseen Chinese character classes.

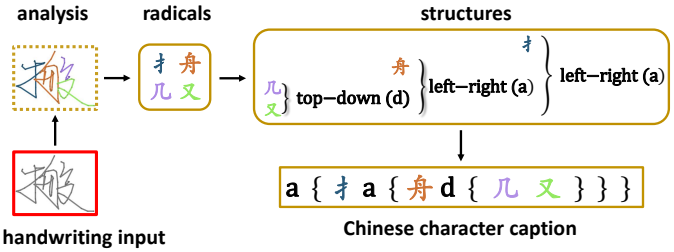

Fig. 1. Illustration of TRAN to recognize Chinese characters by analyzing the radicals and the corresponding structures.

In this paper, we propose a novel radical-based approach to online handwritten Chinese character recognition, namely trajectory-based radical analysis network (TRAN). Different from above mentioned radical-based approaches, in TRAN the radical segmentation and structure detection are automatically addressed by an attention model which is jointly optimized with the entire network. The main idea of TRAN is to decompose a Chinese character into radicals and detect the spatial structures among radicals. We then describe the analysis of radicals as a Chinese character caption. A handwritten Chinese character is successfully recognized when its caption matches ground-truth. To be more accessible, we illustrate the TRAN learning way in Fig. 1. The online handwritten Chinese character input is visualized at the bottom-left of Fig. 1. It is composed of four different radicals. The handwriting input is finally recognized as the bottom-right Chinese character caption after the top-down and left-right structures among radicals are detected. Based on analysis of radicals, the proposed TRAN possesses the capability of recognizing unseen Chinese character classes if the radicals have been seen.

The proposed TRAN is an improved version of attention-

based encoder-decoder model [11] with RNN [12]. The attention-based encoder-decoder model has been extensively applied to many applications including machine translation [13], [14], image captioning [15], [16], speech recognition [17] and mathematical expression recognition [18], [19]. The raw data of online handwritten Chinese character input are variable-length sequence (xy-coordinates). TRAN first employs a stack of bidirectional RNN [20] to encode input sequence into high-level representations. Then a unidirectional RNN decoder converts the high-level representations into output character captions one symbol at a time. For each predicted radical, a coverage based attention model [21] built in the decoder scans the entire input sequence and chooses the most relevant part to describe a segmented radical or a two-dimensional structure between radicals. Our proposed TRAN is related to our previous work [22] with two main differences: 1) [22] focused on the application of RAN on printed Chinese character recognition while this paper focuses on handwritten Chinese character recognition. It is interesting to investigate the performance of RAN on handwritten Chinese character recognition as handwritten characters are much more ambiguous due to the diversity of writing styles. 2) Instead of transforming online handwritten characters into static images and employing convolutional neural network [23] to encode them, we choose to directly encode the raw sequential data by employing an RNN encoder in order to fully exploit the dynamic trajectory information that can not be recovered from static images.

The main contributions of this study are as follows:

- We propose TRAN for online handwritten Chinese character recognition.
- The size of radical vocabulary is largely less than Chinese character vocabulary, leading to decrease of redundancy among output classes and improvement of recognition performance.
- TRAN possess the ability of recognizing unseen or newly created Chinese characters, only if the radicals have been seen.
- We experimentally demonstrate how RAN performs on online handwritten Chinese character recognition compared with state-of-the-arts and show its effectiveness on recognizing unseen character classes.

II. Description of Chinese character caption

In this section, we will introduce how we generate captions of Chinese characters. The character caption is composed of three key components: radicals, spatial structures and a pair of braces (e.g. "{" and "}"). A radical represents a basic part of Chinese character and it is often shared by different Chinese characters. Compared with enormous Chinese character categories, the amount of radicals is quite limited. It is declared in GB13000.1 standard published by National Language Committee of China that nearly 500 radicals consist of over 20,000 Chinese characters. As for the complicated two-dimensional spatial structures among radicals, Fig. 2 illustrates

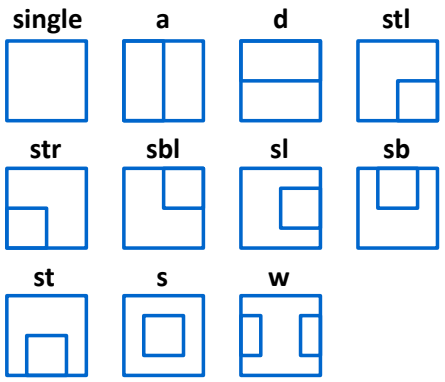

Fig. 2. Graphical representation of eleven common spatial structures among radicals, different radicals are divided by internal line.

eleven common structures and the descriptions are demonstrated as follows: **single-element:** sometimes a single radical represents a Chinese character and therefore we can not find internal structures in such characters, **a:** left-right structure, **d:** top-bottom structure, **stl:** top-left-surround structure, **str:** top-right-surround structure, **sbl:** bottom-left-surround structure, **sl:** left-surround structure, **sb:** bottom-surround structure, **st:** top-surround structure, **s:** surround structure, **w:** within structure

After decomposing Chinese characters into radicals and internal spatial structures by following cjk-decomp [1], we use a pair of braces to constrain a single structure. Take "stl" as an example, it is captioned as "stl { radical-1 radical-2 }". The generation of a Chinese character caption is finished when all radicals are included in the caption.

III. The proposed approach

In this section, we elaborate the proposed TRAN framework, namely generating an underlying Chinese character caption from a sequence of online handwritten trajectory points, as illustrated in Fig. 3. First, we extract trajectory information as the input feature from original trajectory points (xy-coordinates). A stack of bidirectional RNNs are then employed as the encoder to transform the input feature into high-level representations. Since the original trajectory points are a variable-length sequence, the extracted high-level representations are also a variable-length sequence. To associate the variable-length representations with variable-length character caption, we generate a fixed-length context vector via weighted summing the high-level representations and a unidirectional RNN decoder uses the fixed-length context vector to generate the character caption one symbol at a time. We introduce an attention model to produce the weighting coefficients so that the context vector can contain only useful trajectory information at each decoding step.

A. Feature extraction

During the data acquisition of online handwritten Chinese character, the pen-tip movements (xy-coordinates) and pen

[1] https://github.com/amake/cjk-decomp

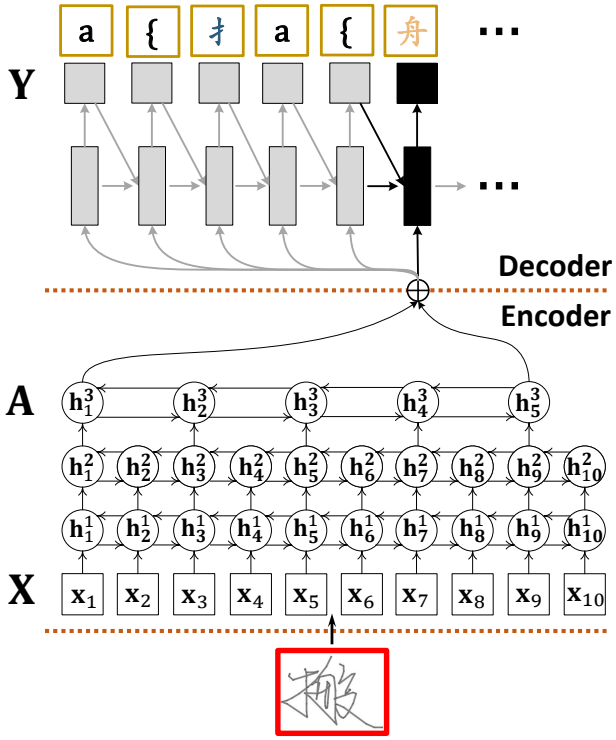

Fig. 3. Overall framework of TRAN for online handwritten Chinese character recognition. It is composed of a bidirectional RNN encoder and a unidirectional RNN decoder.

states (pen-down or pen-up) are stored as variable-length sequential data:

$$\{[x_1, y_1, s_1],\ [x_2, y_2, s_2],\ \ldots,\ [x_N, y_N, s_N]\} \tag{1}$$

where N is the length of sequence, x_i and y_i are the xy-coordinates of the pen movements and s_i indicates which stroke the i^{th} point belongs to.

To address the issue of non-uniform sampling by different writing speed and the size variations of the coordinates on different potable devices, the interpolation and normalization to the original trajectory points are first conducted according to [4]. Then we extract a 6-dimensional feature vector for each point:

$$[x_i, y_i, \Delta x_i, \Delta y_i, \delta(s_i = s_{i+1}), \delta(s_i \neq s_{i+1})] \tag{2}$$

where $\Delta x_i = x_{i+1} - x_i$, $\Delta y_i = y_{i+1} - y_i$, and $\delta(\cdot) = 1$ when the condition is true or zero otherwise. The last two terms are flags which indicate the status of the pen, i.e., $[1,\ 0]$ and $[0,\ 1]$ are pen-down and pen-up respectively. For convenience, in the following sections, we use $\mathbf{X} = (\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_N)$ to denote the input sequence of the encoder, where $\mathbf{x}_i \in \mathbb{R}^d$ $(d = 6)$.

B. Encoder

Given the feature sequence $(\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_N)$, we employ RNN as the encoder to encode them into high-level representations as RNN has shown its strength in processing sequential signals. However, a simple RNN has revealed serious problems during training namely vanishing gradient and exploding gradient [24], [25]. Therefore, an improved version of RNN named gated recurrent units (GRU) [26] which can alleviate these two problems is employed in this study as it utilizes an update gate and a reset gate to control the flow of forward information and backward gradient. The GRU hidden state $\mathbf{h}_t$ in encoder is computed by:

$$\mathbf{h}_t = \text{GRU}\left(\mathbf{x}_t, \mathbf{h}_{t-1}\right) \tag{3}$$

and the GRU function can be expanded as follows:

$$\mathbf{z}_t = \sigma(\mathbf{W}_{xz}\mathbf{x}_t + \mathbf{U}_{hz}\mathbf{h}_{t-1}) \tag{4}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{U}_{hr}\mathbf{h}_{t-1}) \tag{5}$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{U}_{rh}(\mathbf{r}_t \otimes \mathbf{h}_{t-1})) \tag{6}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \otimes \mathbf{h}_{t-1} + \mathbf{z}_t \otimes \tilde{\mathbf{h}}_t \tag{7}$$

where σ denotes the sigmoid activation function, $\otimes$ denotes an element-wise multiplication operator, $\mathbf{z}_t$, $\mathbf{r}_t$ and $\tilde{\mathbf{h}}_t$ are the update gate, reset gate and candidate activation, respectively. $\mathbf{W}_{xz}$, $\mathbf{W}_{xr}$, $\mathbf{W}_{xh}$, $\mathbf{U}_{hz}$, $\mathbf{U}_{hr}$ and $\mathbf{U}_{rh}$ are related weight matrices.

Nevertheless, even if the unidirectional GRU can have access to the history of input signals, it does not have the ability of modeling future context. Therefore we exploit the bidirectional GRU by passing the input vectors through two GRU layers running in opposite directions and concatenating their hidden state vectors so that the encoder can use both history and future information. To obtain a high-level representation, the encoder stacks multiple GRU layers on top of each other as illustrated in Fig. 3. In this study, our encoder consists of 4 bidirectional GRU layers. Each layer has 250 forward and 250 backward GRU units. We also add pooling over time axes in high-level GRU layers because: 1) the high-level representations are overly precise and contain much redundant information; 2) the decoder needs to attend less if the number of encoder output reduces, leading to improvement of performance; 3) the pooling operation accelerates the encoding process. The pooling is applied to the top GRU layer by dropping the even output over time.

Assuming the bidirectional GRU encoder produces a high-level representation sequence $\mathbf{A}$ with length L. Because there is one pooling operation in the bidirectional GRU encoder, $L = \frac{N}{2}$. Each of these representations is a D-dimensional vector $(D = 500)$:

$$\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_L\}\ ,\ \mathbf{a}_i \in \mathbb{R}^D \tag{8}$$

C. Decoder with attention model

After obtaining high-level representations $\mathbf{A}$, the decoder aims to make use of them to generate a Chinese character caption. The output sequence $\mathbf{Y}$ is represented as a sequence of one-hot encoded vectors:

$$\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_C\}\ ,\ \mathbf{y}_i \in \mathbb{R}^K \tag{9}$$

where K is the vocabulary size and C is the length of character caption. Note that, both the length of representation sequence (L) and the length of character caption (C) are variable.

To address the mapping from variable-length representation sequence to variable-length character caption, we attempt to compute an intermediate fixed-size vector $\mathbf{c}_t$ that incorporates useful information of representation sequence. The decoder then utilizes this fixed-size vector to predict the character caption one symbol at a time. As $\mathbf{c}_t$ contains overall information of input sequence, we call it context vector. At each decoding step, the probability of the predicted word is computed by the context vector $\mathbf{c}_t$, current decoder state $\mathbf{s}_t$ and previous predicted symbol $\mathbf{y}_{t-1}$ using a multi-layer perceptron:

$$p(\mathbf{y}_t|\mathbf{y}_{t-1}, \mathbf{X}) = g\left(\mathbf{W}_o h(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{W}_s \mathbf{s}_t + \mathbf{W}_c \mathbf{c}_t)\right) \quad (10)$$

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

Since the context vector $\mathbf{c}_t$ needs to be fixed-length, it is an intuitive way to produce it by summing all representation vectors $\mathbf{a}_i$ at time step t. However, average summing is too robust and leads to loss of useful information. Therefore, we adopt weighted summing while the weighting coefficients are called attention probabilities. The attention probability performs as a description that tells which part of representation sequence is useful at each decoding step. We compute the decoder state $\mathbf{s}_t$ and context vector $\mathbf{c}_t$ as follows:

$$\hat{\mathbf{s}}_t = \text{GRU}\left(\mathbf{y}_{t-1}, \mathbf{s}_{t-1}\right) \quad (11)$$

$$\mathbf{F} = \mathbf{Q} * \sum\nolimits_{l=1}^{t-1} \boldsymbol{\alpha}_l \quad (12)$$

$$e_{ti} = \boldsymbol{\nu}_{\text{att}}^{\text{T}} \tanh(\mathbf{W}_{\text{att}} \hat{\mathbf{s}}_t + \mathbf{U}_{\text{att}} \mathbf{a}_i + \mathbf{U}_f \mathbf{f}_i) \quad (13)$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})} \quad (14)$$

$$\mathbf{c}_t = \sum\nolimits_{i=1}^{L} \alpha_{ti} \mathbf{a}_i \quad (15)$$

$$\mathbf{s}_t = \text{GRU}\left(\mathbf{c}_t, \hat{\mathbf{s}}_t\right) \quad (16)$$

Here, we can see that the decoder adopts two unidirectional GRU layers to calculate the decoder state $\mathbf{s}_t$. The GRU function is the same one in Eq. (3). $\hat{\mathbf{s}}_t$ denotes the current decoder state prediction, e_{ti} denotes the energy of $\mathbf{a}_i$ at time step t conditioned on $\hat{\mathbf{s}}_t$. The attention probability α_{ti}, which is the i^{th} element of $\boldsymbol{\alpha}_t$, is computed by taking e_{ti} as input of a softmax function. The context vector $\mathbf{c}_t$ is then calculated via weighted summing representation vectors $\mathbf{a}_i$ with attention probabilities employed as weighting coefficients. During the computation of attention probability, we also append a coverage vector $\mathbf{f}_i$ (the i^{th} vector of $\mathbf{F}$) in the attention model. The coverage vector is computed based on the summation of all past attention probabilities so that the coverage vector contains the information of alignment history as shown in Eq. (12). We adopt the coverage vector in order to let the attention model know which part of representation sequence has been attended or not [27]. Let $n^{'}$ denote the attention dimension. Then $\boldsymbol{\nu}_{\text{att}} \in \mathbb{R}^{n^{'}}$, $\mathbf{W}_{\text{att}} \in \mathbb{R}^{n^{'} \times n}$ and $\mathbf{U}_{\text{att}} \in \mathbb{R}^{n^{'} \times D}$.

IV. Training and Testing Details

The training objective of the proposed model is to maximize the predicted symbol probability as shown in Eq. (10) and we use cross-entropy (CE) as the objective function:

$$O = -\sum\nolimits_{t=1}^{C} \log p(w_t|\mathbf{y}_{t-1}, \mathbf{X}) \quad (17)$$

where w_t represents the ground truth word at time step t, C is the length of output string. The implementation details of GRU encoder has been introduced in Section III-B. The decoder uses two layers with each using 256 forward GRU units. The embedding dimension m, decoder state dimension n and attention dimension n' are all set to 256. The convolution kernel size for computing coverage vector is set to (5×1) as it is a one-dimensional convolution operation, while the number of convolution filters is set to 256. We utilize the adadelta algorithm [28] with gradient clipping for optimization. The adadelta hyperparameters are set as $\rho = 0.95$, $\varepsilon = 10^{-8}$.

In the decoding stage, we aim to generate a most likely character caption given the input trajectory:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \log P\left(\mathbf{y}|\mathbf{X}\right) \quad (18)$$

However, different from the training procedure, we do not have the ground truth of previous predicted word. To prevent previous prediction errors inherited by next decoding step , a simple left-to-right beam search algorithm [29] is employed to implement the decoding procedure. Here, we maintained a set of 10 partial hypotheses beginning with the start-of-sentence $< sos >$. At each time step, each partial hypothesis in the beam is expanded with every possible word and only the 10 most likely beams are kept. This procedure is repeated until the output word becomes the end-of-sentence $< eos >$.

V. Experiments

In this section, we present experiments on recognizing seen and unseen online handwritten Chinese character classes by answering the following questions:

- Q1 Is the TRAN effective when recognizing seen Chinese character classes?
- Q2 Is the TRAN effective when recognizing unseen Chinese character classes?
- Q3 How does the TRAN analyze the radicals and spatial structures?

A. Performance on recognition of seen Chinese character classes (Q1)

In this section, we show the effectiveness of TRAN on recognizing seen Chinese character classes. The set of character class is 3,755 commonly used Chinese characters. The dataset used for training is the CASIA [30] dataset including OLHWDB1.0 and OLHWDB1.1. There are totally 2,693,183 samples for training and 224,590 samples for testing. The training and testing data were produced by different writers with enormous handwriting styles across individuals. In Table I, the human performance on CASIA test set and the previous benchmark are both listed. NET4 is the proposed

TABLE I
RESULTS ON CASIA DATASET OF ONLINE HANDWRITTEN CHINESE CHARACTER RECOGNITION.

Methods	Reference	Accuracy
Human Performance	[31]	95.19%
Traditional Benchmark	[32]	95.31%
NET4	[4]	96.03%
TRAN	–	96.43%

method in [4] which represents the state-of-the-art method on CASIA dataset and it belongs to non-radical based methods. NET4 achieved an accuracy of 96.03% while TRAN achieved an accuracy of 96.43%, revealing relative character error rate reduction of 10%. To be fairly comparable, here NET4 and TRAN both did not use the sequential dropout trick as proposed in [4] so the performance of NET4 is not as good as the best performance presented in [4]. As explained in the contributions of this study in Section I, the main difference between radical based method and non-radical based method for Chinese character recognition is the size of radical vocabulary is largely less than Chinese character vocabulary, yielding decrease of redundancy among output classes and improvement of recognition performance.

B. Performance on recognition of unseen Chinese character classes (Q2)

The number of Chinese character classes is not fixed as more and more novel characters are being created. Also, the overall Chinese character classes are enormous and it is difficult to train a recognition system that covers them all. Therefore it is necessary for a recognition system to possess the capability of recognizing unseen Chinese characters, called zero-shot learning.

Obviously traditional non-radical based methods are incapable of recognizing these unseen characters since the objective character class has never been seen during training procedure. However TRAN is able to recognize unseen Chinese characters only if the radicals composing unseen characters have been seen. To validate the performance of TRAN on recognizing unseen Chinese character classes, we divide 3755 common Chinese characters into 3255 classes and the other 500 classes. We choose handwritten characters belonging to 3255 classes from original training set as the new training set and we choose handwritten characters belonging to the other 500 classes from original testing set as the new testing set. By doing so, both the testing character classes and handwriting variations have never been seen during training. We explore different size of training set to train TRAN, ranging from 500 to 3255 Chinese character classes and we make sure the radicals of testing characters are covered in training set.

We can see in Table II the recognition accuracy of unseen Chinese character classes is not available when training set only contains 500 Chinese character classes. We believe it is difficult to train TRAN properly to accommodate large handwriting variations when the number of character classes

TABLE II
RESULTS ON NEWLY DIVIDED TESTING SET BASED ON CASIA DATASET OF ONLINE HANDWRITTEN UNSEEN CHINESE CHARACTER RECOGNITION, TESTING SET CONTAINS 500 CHINESE CHARACTER CLASSES.

Train classes	Train samples	Test Accuracy
500	359,036	-
1000	717,194	10.74%
1500	1,075,344	26.02%
2000	1,435,295	39.35%
2755	1,975,972	50.45%
3255	2,335,433	60.37%

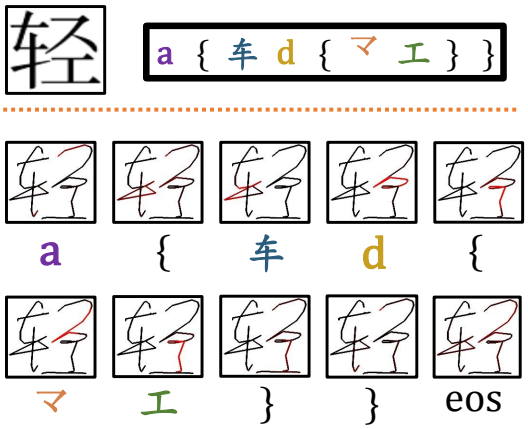

Fig. 4. Examples of attention visualization during the decoding procedure. The red color on trajectory describes the attention probabilities namely the lighter color denotes higher attention probabilities and the darker color denotes lower attention probabilities.

is quite small. When the training set contains 3255 character classes, TRAN achieves a character accuracy of **60.37**% which is a relatively pleasant performance compared with traditional recognition systems as they can not recognize unseen Chinese character classes which means their accuracies are definitely **0**%. The performance of recognizing unseen Chinese character classes is not as good as the performance presented in [22] because the handwritten Chinese characters are much more ambiguous compared with printed Chinese characters due to the large handwriting variations.

C. Attention visualization (Q3)

In this section, we show through attention visualization how TRAN is able to recognize internal radicals and analyze the two-dimensional spatial structure among radicals. Fig. 4 illustrates an example of attention visualization. Above the dotted line, there is one Chinese character class and its corresponding character caption. Below dotted line, there are images denoting the visualization of attention probabilities during decoding procedure. We draw the trajectory of input handwritten Chinese character in a two-dimensional greyscale image to visualize attention. Below images there are corresponding symbols generated by decoder at each decoding step.

As we can see in Fig. 4, when encountering basic radicals, the attention model generates the alignment well correspond-

ing to the human intuition. Also, it mainly focus on the ending of last radical and the beginning of next radical to detect a spatial structure. Take "d" as an example, by attending to the ending of last radical and the beginning of next radical, the attention model detects a top-bottom direction, therefore a top-bottom structure is analyzed. Immediately after generating a spatial structure, the decoder produces a pair of braces "{}", which are employed to constrain the two-dimensional structure in Chinese character caption.

VI. Conclusion

In this study we introduce TRAN for online handwritten Chinese character recognition. The proposed TRAN recognizes Chinese character by identifying internal radicals and analyzing spatial structures among radicals. We show from experimental results that TRAN outperforms the state-of-the-art method on recognition of online handwritten Chinese characters and possesses the capability of recognizing unseen Chinese character categories. By visualizing learned attention probabilities, we can observe the alignments of radicals and analysis of structures correspond well to human intuition.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under contract No. 2017YFB1002202, the National Natural Science Foundation of China under Grants No. 61671422 and U1613211, the Key Science and Technology Project of Anhui Province under Grant No. 17030901005, and MOE-Microsoft Key Laboratory of USTC.

References

[1] C. Y. Suen, M. Berthod, and S. Mori, "Automatic recognition of handprinted characters–the state of the art," *Proceedings of the IEEE*, vol. 68, no. 4, pp. 469–487, 1980.

[2] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[3] C.-L. Liu, S. Jaeger, and M. Nakagawa, "Online recognition of Chinese characters: the state-of-the-art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 2, pp. 198–213, 2004.

[4] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing Chinese characters with recurrent neural network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[5] W. Yang, L. Jin, D. Tao, Z. Xie, and Z. Feng, "Dropsample: A new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten chinese character recognition," *Pattern Recognition*, vol. 58, pp. 190–203, 2016.

[6] Z. Zhong, L. Jin, and Z. Xie, "High performance offline handwritten chinese character recognition using googlenet and directional feature maps," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 846–850.

[7] S.-k. Chang, "An interactive system for Chinese character generation and retrieval," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 3, pp. 257–265, 1973.

[8] A.-B. Wang and K.-C. Fan, "Optical recognition of handwritten Chinese characters by hierarchical radical matching method," *Pattern Recognition*, vol. 34, no. 1, pp. 15–35, 2001.

[9] L.-L. Ma and C.-L. Liu, "A new radical-based approach to online handwritten Chinese character recognition," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[10] T.-Q. Wang, F. Yin, and C.-L. Liu, "Radical-based Chinese character recognition via multi-labeled learning of deep residual networks," in *Document Analysis and Recognition (ICDAR), 2017 14th International Conference on*. IEEE, 2017.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[12] A. Graves *et al.*, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, vol. 385.

[13] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[14] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[15] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[16] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[17] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.

[18] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition*, 2017.

[19] J. Zhang, J. Du, and L. Dai, "Multi-scale attention with dense encoder for handwritten mathematical expression recognition," *arXiv preprint arXiv:1801.03530*, 2018.

[20] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[21] J. Zhang, J. Du, and L. Dai, "A gru-based encoder-decoder approach with attention for online handwritten mathematical expression recognition," in *Document Analysis and Recognition (ICDAR), 2017 14th International Conference on*. IEEE, 2017.

[22] J. Zhang, Y. Zhu, J. Du, and L. Dai, "Radical analysis network for zero-shot learning in printed Chinese character recognition," *arXiv preprint arXiv:1711.01889*, 2017.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[24] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[25] J. Zhang, J. Tang, and L.-R. Dai, "RNN-BLSTM based multi-pitch estimation." in *INTERSPEECH*, 2016, pp. 1785–1789.

[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[27] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," *arXiv preprint arXiv:1601.04811*, 2016.

[28] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[29] K. Cho, "Natural language understanding with distributed representation," *arXiv preprint arXiv:1511.07916*, 2015.

[30] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "CASIA online and offline Chinese handwriting databases," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 37–41.

[31] F. Yin, Q.-F. Wang, X.-Y. Zhang, and C.-L. Liu, "ICDAR 2013 Chinese handwriting recognition competition," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1464–1470.

[32] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "Online and offline handwritten Chinese character recognition: benchmarking on new databases," *Pattern Recognition*, vol. 46, no. 1, pp. 155–162, 2013.