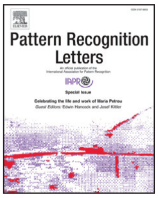
Pattern Recognition Letters 125 (2019) 821–827

Contents lists available at ScienceDirect

Pattern Recognition Letters

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



Radical aggregation network for few-shot offline handwritten Chinese character recognition

Tianwei Wang, Zecheng Xie, Zhe Li, Lianwen Jin[*], Xiangle Chen

School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China

ARTICLE INFO

Article history:
Received 22 May 2019
Revised 2 August 2019
Accepted 3 August 2019
Available online 4 August 2019

Keywords:
Handwritten Chinese character recognition
Chinese radical recognition
Deep learning
Few-shot learning

ABSTRACT

Offline handwritten Chinese character recognition has attracted much interest due to its various applications. The most cutting-edge methods treat Chinese character as a whole, ignoring the structures and radicals that compose characters. To use the radical-level composition of Chinese characters and achieve few-shot/zero-shot Chinese character recognition, some methods attempt to recognize Chinese characters at the radical level; however, these methods have shown poor performance due to weak radical feature representation and the use of inflexible decoding algorithm. In this paper, a novel radical aggregation network (RAN) is proposed for few-shot/zero-shot offline handwritten Chinese character recognition. The RAN consists of three components, a radical mapping encoder (RME), a radical aggregation module (RAM), and a character analysis decoder (CAD). Experiments show that our method can effectively recognize unseen handwritten characters given few support samples, while maintaining a high performance on seen characters.

1. Introduction

Offline Handwritten Chinese characters recognition (OHCCR) has been studied for more than 40 years; however, it remains a challenge due to the size of the vocabulary (70224 in GB18010-2005), diversity in writing styles, similarity in characters, and so on. Early works on handwritten Chinese character recognition often rely on hand-crafted features [20,21,27,28]. In recent years however, HCCR has experienced a rapid evolution [4,8,9,25,33] owing to recent developments in deep learning and convolutional neural network (CNN). The multi-column deep neural network method proposed by Cireşan et al. [4], may be the first method in which a CNN was successfully applied for large vocabulary HCCR; it showed better performance than traditional methods. Li et al. [9] proposed a CNN-based method that outperformed even humans. Xiao et al. [25] implemented an iterative refinement strategy in the recognizer for fine-grained recognition of confusing characters. However, all of these methods treat recognition at the character level, disregarding the basic parts that comprise Chinese characters, i.e., radicals [14,15]. In many cases, handwritten samples are expensive to obtain. Previous methods may lack flexibility because they could only recognize seen characters.

In this paper, we focus on radical-based few-shot or zero-shot OHCCR, where zero-shot refers to the classifier which is trained with limited Chinese character classes, containing all radicals; and the recognition is then performed on the unseen classes. Few-shot means that few support samples of the unseen classes are also added for training. Compared to the large-scale and ever-increasing characters in Chinese languages, approximately 1000 radicals can be used to compose over 10,000 characters [23]. All of Chinese characters can be decomposed into a unique radical string (Fig. 1), e.g., character “辉” can be decomposed as {“光”,“冖”,“车”}. When people learn Chinese characters, they first learn the radicals and structures that form characters. By learning radicals and structures, the difficulty of learning to read and write Chinese characters reduces significantly. Similarly, the recognition of character can be decomposed into recognizing radicals and structures. Solving OHCCR at the radical level can significantly reduce training sample dependency.

Despite the huge number of Chinese characters, current state-of-the-art methods [24,30] achieved radical-based Chinese character zero-shot recognition with an encoder-decoder architecture. They proposed to use a convolution-based encoder to extract visual features, and use an attention-based decoder to generate radical captions. Finally they matched the radical captions with pre-defined character decomposition strings. Although the aforementioned methods can recognize unseen characters, they perform

[*] Corresponding author.
E-mail address: eelwjin@scut.edu.cn (L. Jin).

https://doi.org/10.1016/j.patrec.2019.08.005

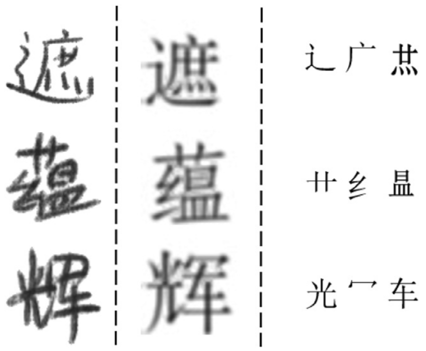


Fig. 1. Handwritten and printed Chinese character samples and their corresponding radicals. Left: handwritten Chinese character. Middle: printed Chinese character. Right: corresponding radicals.

poorly for two main reasons. The first is weak radical feature representation: Previous methods generate radical captions by performing linear classification with the softmax function on the radical features, which is a discriminative model that ignores the feature distribution. The second reason is that the previous methods perform prediction by match decoding, where the final decision is made by matching the radical recognition result with the pre-defined character-radical lexicon. The performance of match decoding depends strongly on the efficiency of the pre-defined lexicon and decoding strategy, which is inflexible.

In this paper, we propose a novel radical-based approach for few-shot/zero-shot OHCCR, called the radical aggregation network (RAN). Our method applies an encoder-decoder architecture, which consists of three components, namely a radical mapping encoder (RME), a radical aggregation module (RAM), and a character analysis decoder (CAD). In summary, our main contributions are summarized as follows:

- For effective and robust radical representation, we introduce the RAM, which combines deep prototype learning for robust radical feature representation. It uses a distance metric criterion for radical feature representation to promote the intra-class compactness, while encouraging inter-class differences.
- To achieve end-to-end decoding, we introduce the CAD, which analyses the radical representation and makes final prediction with few support samples, thus avoiding the use of inflexible match decoding.
- We experimentally demonstrate that our approach can effectively recognize unseen handwritten characters while maintaining a comparable performance for seen characters.

2. Related work

Early work on OHCCR focused more on the design of visual features for classification models [21,27,28]. With the rapid development of deep convolutional neural network, a number of CNN-based models have been proposed and have achieved great performance improvements in a variety of fields. Ciresan et al. [4] first introduced the multi-column convolutional neural network to HCCR, and achieved performance comparable to that of humans. Zhong et al. [33] used streamlined GoogleNet [19] for HCCR and outperformed humans. To address the problem of expensive data labeling, Liu et al. [12] proposed a simple and effective method, called gate-guided dynamic learning to automatically label large amounts of handwritten data. Aiming at recognizing the similar characters, Xiao et al. [25] presented an iterative attention mechanism to utilize both low-level visual cues and high-level information. Zhang et al. [32] proposed to extract writer-independent semantic features from handwritten characters by incorporating the prior knowledge of printed data and writer-independent semantic features to improve the performance of HCCR. All these methods address HCCR at the character level; thus, they cannot recognize unseen characters.

Most Chinese characters can be divided into a sequence of radicals. Addressing HCCR at the radical level can reduce data dependency. Wang et al. [22] proposed a method that extracted radials and matched the lexicon with a hierarchical radical matching scheme. Ma et al. [14] proposed to over-segment characters to radicals and then find the corresponding character in the lexicon; however, this method could only handle a left-right structure. Wang et al. [23] addressed radical recognition using a multi-label learning using deep residual network; however, they ignored the structure information. Zhang et al. [30] and Wang et al. [24] proposed radical analysis networks having an encoder-decoder architecture, in which the encoder extracts visual features and the attention-based decoder generates radical captions.

Few-shot classification aims to learn a classifier to recognize unseen classes during training with limited labeled examples [2]. Recently, distance metric learning based few-shot learning method has attracted attention. This method addresses few-shot classification by calculating and comparing distances in a high-dimensional feature space. Snell et al. [17] proposed learning a metric space in which classification can be performed by computing Euclidean distances to prototype representations for each class, this approach proved to be simple and effective. Long et al. [13] also proposed to achieve zero-shot learning by forcing samples to be intra-class aggregation and inter-class separation. Prototype learning is a classical topic in pattern recognition [10]. The prototype is a representation of each class in the feature space. Yang et al. [26] has shown that prototype learning can improve the robustness of features.

Different from [17], in which the mean vector of the same class is computed as the prototype, we randomly initialized prototypes of each class and optimized them with a back propagation algorithm [16].

3. Method

The overall architecture of our method is shown in Fig. 2. The radical mapping encoder (RME) maps the input image into a radical representation sequence, of which each representation is a high-dimensional feature vector. The radical aggregation module (RAM) conducts a distance metric between the radical representation and radical prototypes; it then aggregates radicals to its own prototypes while distancing it from others. The character analysis decoder (CAD) analyses the radical representations sequentially and transcripts them into character.

3.1. Radical mapping encoder

Supposed a Chinese character has L radicals, for example, character “辉” consists of radical “光”,“冖”,“车”, $L = 3$. The RME maps the input into radical representation sequence $\boldsymbol{r} = \{r_1, r_2, \ldots, r_{L+1}\}$ (+1 for EOS token [18]), r_t is a high-dimensional vector representing radical in the feature space. The RME first extracts the visual features from input image using a residual convolutional neural network (ResNet) [5]. Then, it generates radical representations sequentially using the attention module [1].

3.1.1. Feature extractor

ResNet has been proven to be an effective feature extractor for many visual tasks. In this paper we use ResNet as our feature extractor to encode a raw input image x to a features map $f(x;\ \theta)$, where θ represents the encoder parameters. As shown

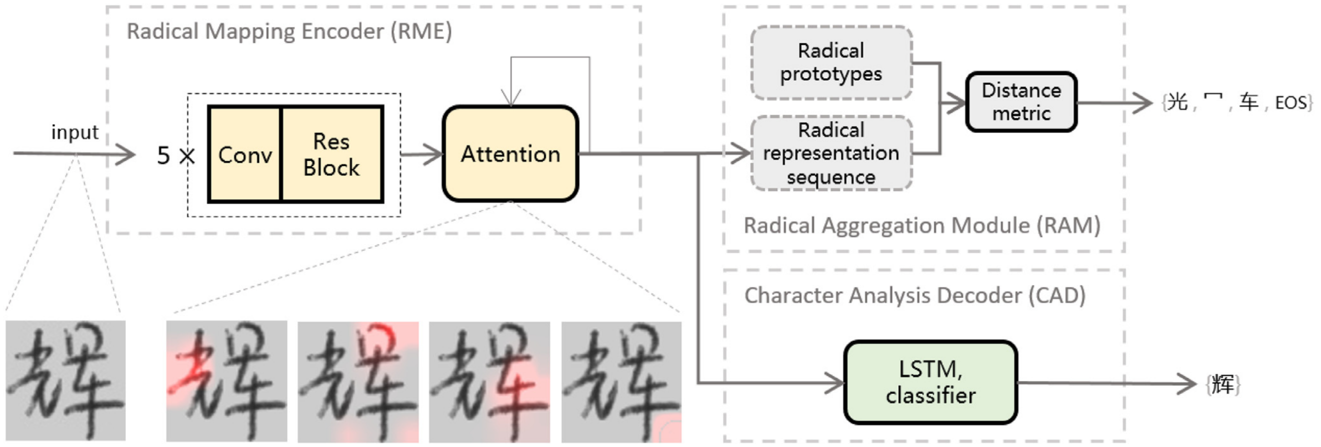

Radical Mapping Encoder (RME)
input
5 ×
Conv
Res Block
Attention
Radical prototypes
Radical representation sequence
Distance metric
{光, 冖, 车, EOS}
Radical Aggregation Module (RAM)
Character Analysis Decoder (CAD)
LSTM, classifier
{辉}

Fig. 2. Overall architecture of our radical aggregation network.

in Fig. 2, the encoder consists of five basic convolution blocks, which can be denoted as:

$$x_l = H(Conv(x_{l-1})) + Conv(x_{l-1}) \tag{1}$$

where x_l denotes feature map in layer *l*, *Conv* is a convolution layer with activate function, and *H* is a residual block.

3.1.2. Attention module

The same as other attention modules [1,24], in which the feature map $f(x;\ \theta)$ is further encoded by a recurrent network, $f(x;\ \theta)$ is flattened by the column then encoded by a bi-directional long short-term memory (LSTM) [6] network, which yields $\boldsymbol{h} = (h_1, h_2, \ldots, h_N)$. The radical representation r_t is the hidden state of the gated recurrent unit (GRU) [3] at time *t*:

$$r_t = GRU(c_t, r_{t-1}) \tag{2}$$

where c_t is the relevant contents computed as the weighted sum of the features:

$$c_t = \sum_{j=1}^{N} \alpha_{t,j} h_j \tag{3}$$

The attention weights $\alpha_{t,j}$ is computed by:

$$\alpha_{t,j} = \frac{exp(e_{t,j})}{\sum_{j=1}^{N} exp(e_{t,j})} \tag{4}$$

where $e_{t,j}$ is the alignment score, which is computed as:

$$e_{t,j} = f_{attn}(r_{t-1}, h_j) \tag{5}$$

The alignment function f_{attn} is parameterized by a multi-layer perceptron such that:

$$f_{attn}(r_{t-1}, h_j) = V_a Tanh(W_s r_{t-1} + W_f h_j + b) \tag{6}$$

r_t can be classified by a linear classifier:

$$y_r = W_o r_t + b_o \tag{7}$$

the probability of radical vector r_t belongs to class *k* can be computed as:

$$p_1(r_t = k; x, \phi_r) = \frac{exp(y_{rk})}{\sum_{k'}^{K} exp(y_{rk'})} \tag{8}$$

where $\phi_r = W_o, b_o, V_a, W_s, W_f, b$ are all the trainable parameters of RME. The loss function of RME is:

$$L_{RME} = -\sum_{t=1}^{T} logp_1(r_t = k; x, \phi_r) \tag{9}$$

The attention module completes the generation when it predicts an end-of-sequence token "*EOS*".

3.2. Radical aggregation module

To achieve end-to-end recognition of the unseen classes, few support samples serve as a guidance of the unseen classes for the recognizer. For better use of support samples, radical representations should remain robust when faced with different samples styles, i.e., different samples of the same radical should share the same feature representation. Hence, we use deep prototype learning, which has proven to be an effective and robust feature learning method in [26]. The RAM jointly learns the radical representations and their corresponding prototypes with distance metric criterion.

Given *K* radicals, their prototypes are denoted as $\boldsymbol{p} = (p_1, p_2, \ldots, p_K)$, each prototype is a trainable high-dimensional vector. We evaluate the distance between r_t and p_k using the squared Euclidean distance function:

$$d(r_t, p_k) = ||r_t - p_k||_2^2 \tag{10}$$

the probability of radical vector r_t belonging to class *k* can also be computed as:

$$p_2(r_t = k; x, \phi_r) = \frac{exp(-d(r_t, p_k))}{\sum_{k'}^{K} exp(-d(r_t, p_{k'}))} \tag{11}$$

The loss function of the RAM is:

$$L_{RAM} = -\sum_{l=1}^{L+1} logp_2(r_l = k; x, \phi) + \beta_1 \sum_{l=1}^{L+1} d(r_l, p_k) \tag{12}$$

where β_1 is a hyper-parameter. The second term is meant to constrain the intra-class distance.

Discussion: The squared Euclidean distance is a simple yet effective feature distance metric, which has been proven to outperform cosine similarity [17], hence, we use it as our distance metric. As shown in Fig. 3, through radical aggregation, the same radicals cluster around corresponding prototypes. Ideally, the same radical from different samples will be mapped onto the same point in feature space. For better visualization, we plot the radical representations with/without aggregation in Fig. 5, the representations have been reduced to 2-D. It can be seen that, though classified into the same class, print samples and handwritten samples cluster separately in the feature space, which is not a robust condition. Through aggregation, the instances of the same class cluster despite the different styles, which is much more robust. The classification of radicals can be simply implemented by finding the nearest prototype.

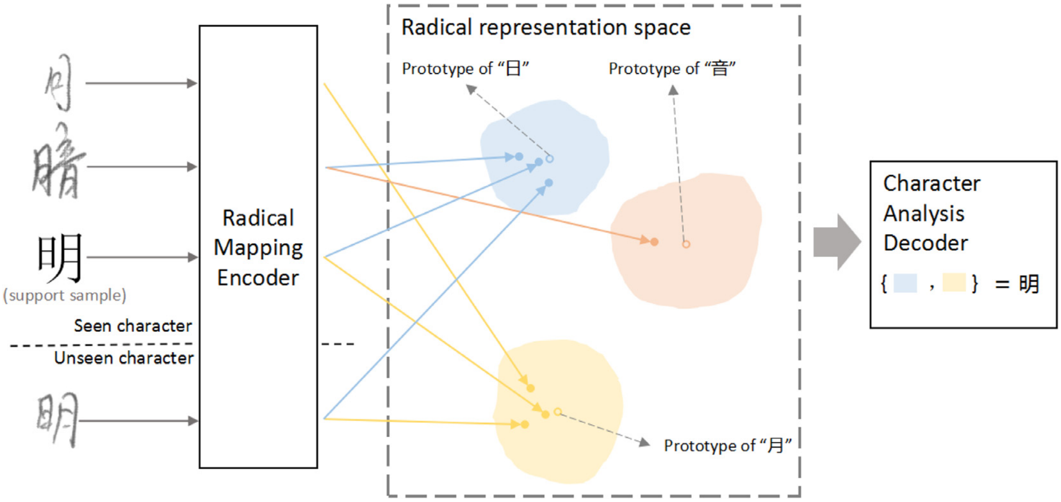


Fig. 3. Process of recognizing an unseen handwritten character. RAN maps radicals into radical representation space and aggregates them; then, unseen characters with the structure information given by the support sample are classified.

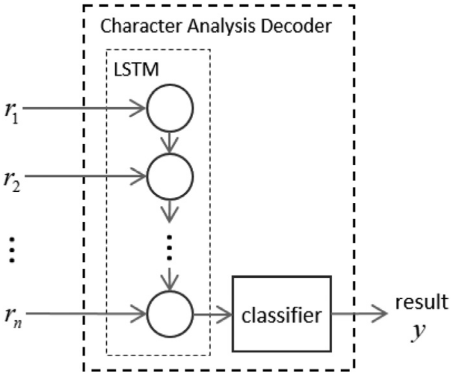


Fig. 4. Architecture of CAD.

3.3. Character analysis decoder

Through RME, we have a radical representation sequence $\boldsymbol{r} = (r_1, r_2, \ldots, r_{L+1})$, which represents the radicals in the input character. We can easily determine exactly what the radicals are by decoding $\boldsymbol{r}$ and then match the result with the character-radical lexicon to make the final decision. This is what we refer to as match decoding. Match decoding always requires some post-processing approaches, such as beam-search to achieve high accuracy [14]; however, such post-processing strategies are complex and inflexible. Thus, we propose the CAD to analyze the radical matrix and generate final classification result.

As shown in Fig. 4, the CAD consists of an LSTM layer and a classifier. The LSTM layer takes $\boldsymbol{r}$ as input sequentially. Its last hidden state h_L can be considered as global information of radical information. Then, the classifier computes the output as follows:

$$y = W_c h_L + b_c \tag{13}$$

the probability of x belongs to class c is:

$$p(x = c; x, \phi) = \frac{exp(y_c)}{\sum_{c'}^{C} exp(y_{c'})} \tag{14}$$

where ϕ is all of the trainable parameters. The loss function of the CAD is computed as:

$$L_{CAD} = -logp(x = k; x, \phi) \tag{15}$$

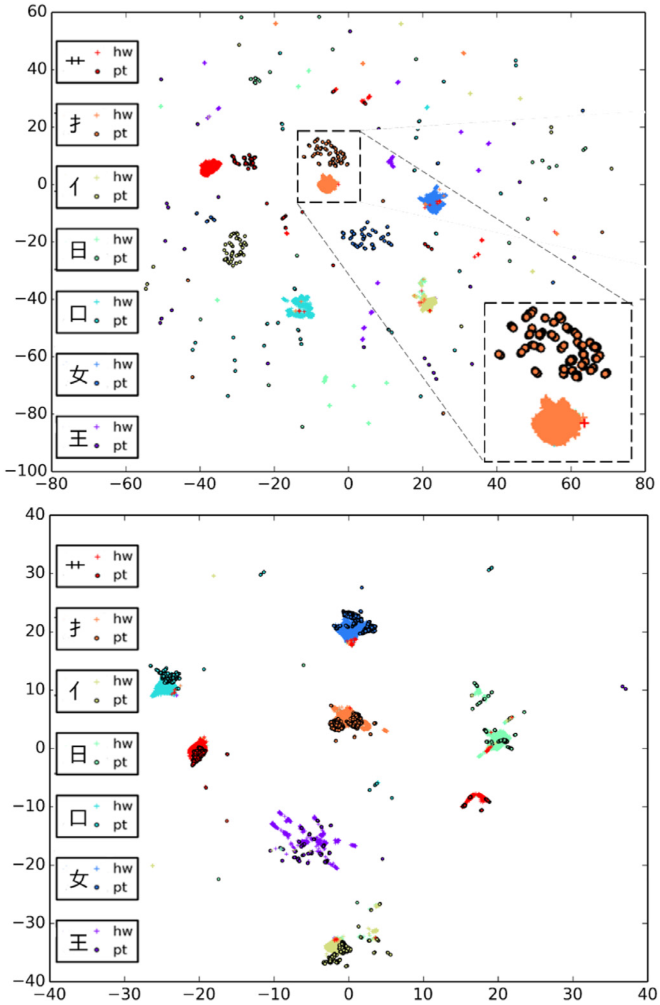


Fig. 5. Visualization of radical features. Top: radical representation without aggregation. Bottom: radical representation with aggregation. We randomly choose 7 radicals, with 500 instances of each radical (half handwritten and half printed). the high-dimensional representations are reduced to 2-D with LargeVis algorithm [7] then plotted.

We jointly optimize RME and CAD as:

$$L = \beta_2 L_{RME} + \beta_3 L_{RAM} + \beta_4 L_{CAD} \tag{16}$$

The hyper-parameters β_2, β_3, and β_4 are meant to balance the three terms.

4. Experiments

In this section, we systematically verify the effectiveness of our method on unseen and seen characters.

Support samples: Compared to handwritten Chinese character samples, printed Chinese character samples are much easier to obtain. Thus, we use printed samples as support data for the unseen handwritten characters. Compared with numerous writing styles in handwritten samples, printed samples show little diversity, which mainly appears between different font styles. The experiment herein uses N fonts for training, this is referred to as N-shot.

We generate support samples with a simple synthetic data generation strategy as follows:

- Print characters to a 64 × 64 white background with *.ttf* font files. The size of the printed character is randomly chosen between 40 to 56 pixel.
- Randomly rotate the image obtained in the previous step by ±2 degrees, then apply 3 × 3 Gaussian blur.

4.1. Experiment settings

A radical is viewed as a part of the semantics and is shared by different characters [15]. Ma et al. [14] extracted 1118 substructures from 4284 characters to build a radical lexicon, Wang et al. [23] extended this lexicon to 9820 characters. Based on the radical extraction strategy, [23] further divide Chinese characters into three main categories: Normal characters, which can be defined by a particular sequence of radicals, like “辉”. Single-structure characters, which can be considered as a radical, like “光”. Multi-structure characters, which has different radicals in different fonts, like “繡”. Each character in this lexicon usually has 1–4 radicals, which means $L \in$ [1,4].

We use CASIA-HWDB 1.0–1.2 [27,28] database and the aforementioned lexicon [22] for evaluation. The database contains 7356 classes of symbols and characters. We filter out the symbols, single-structure, and multistructure characters in the database because of their ambiguity and character-to-radical transcription, leaving 6391 normal characters with 865 radicals for experiments.

We randomly divide these 6391 characters into two sets, set A and set B. Set A has M($0 \leq M \leq 6391$) classes of characters with all of the 865 radicals, each character contains 200 samples. Set B has the other (6391 - M) classes of characters and each character contains 50 samples. Set A is used as a training set and set B is used as the testing set. When only using set A as the training set, we call it zero-shot because none of the characters in set B have appeared during training. Meanwhile, the printed support dataset is denoted as support set C; it contains all of the 6391 characters with N font styles. When jointly using set A and set C as training sets, it becomes N-shot learning.

Implementation Details: All the kernels of feature extractor are set as 3 × 3, and the channels of feature extractor are set as 24, 36, 64, 128, and 256 sequentially. The size of input image x is normalized to $64 \times 64 \times 1$. The first two dimensions refers to height and width, while the third dimensional refers to grayscale. The size of $f(x;\,\theta)$ is $6 \times 6 \times 256$. All layers in the CAD are set to 512 channels. The prototype is a 512-dimensional vector, with all of the elements randomly initialized to $(-1, 1)$. All of the experiments are conducted with the ADADELTA [29] optimization method, with the learning rate set as 1 constantly and β_1 in Eq. (12) set as 0.2. In each minibatch the samples from set A and set C have the same occurrence probability if set C is used.

Table 1
Performance comparison among training strategies.

	Baseline model	(a)	(b)	(c)
Support samples	✓		✓	✓
Radical aggregation				✓
Match decoding	/	35.3%	70.2%	78.2%
E2E decoding	21.0%	/	73.8%	85.8%
Forward speed	1.97 ms/image	4.12 ms/image (Match) 4.34 ms/image (E2E)		

4.2. Ablation study

In this subsection, we set M to a constant 3500 and set N as 10, i.e., 3500 classes of handwritten characters are used for training and set C has 10 font styles.

To verify the effect of general CNN-based character classifier in this case, we design a simple baseline model, which is a character classifier taking the feature extractor in the RME as the backbone.

Note that the we use two decoding strategies: match decoding and end-to-end decoding (E2E decoding). Under match decoding, the result is considered as correct only when the radical sequence completely matches the radical lexicon. In the zero-shot experiment, only match decoding is used.

The experiment results are shown in Table 1. The baseline model achieves a low accuracy of 21.0%, which reflects that the general CNN-based classifier cannot handle the few-shot problem. By comparing exp (a) and exp (b), we can conclude that the printed support samples significantly improve the performance of recognizing unseen characters. Through radical aggregation, the accuracy increases by 8.0% in match decoding and 12.1% in end-to-end decoding, which proves the effect of the proposed radical aggregation. Furthermore, E2E decoding outperforms match decoding as in exp (b) (c), especially in exp (c) using radical aggregation, which means that robust representation helps to learn an end-to-end classifier.

There are two possible reasons to explain this superiority in performance: When using end-to-end decoding, the problem of composing radicals into characters can be considered as a mapping between vectors, which is much softer than matching the radical sequence with the lexicon. End-to-end decoding provides character level supervision to train the model, it introduces a clear connection order information for radical representations in the feature space.

RAN takes more than twice the time of baseline model; the additional computation is mainly caused by the recurrent attention module. But considering that both speeds are in the millisecond level, the additional time consuming is acceptable. End-to-end decoding takes more time than match decoding because it contains an extra CAD module.

4.3. Experiments on recognition of unseen characters

In this subsection, we explore the influence of different values of M and N.

To explore the effect of different values of N, i.e., the font styles used in support set, we fix M as 3500, and set N as 0, 1, 3, 5, 10, and 30 sequentially. The experimental results are shown in Fig. 6, the accuracy appears to rise and then stabilize. When N is less than 10, the accuracy increases as more font styles are added. When N reaches approximately 10, the accuracy reaches a limit of approximately 86%. Meanwhile, E2E decoding stably exceeds match decoding by 5%–8%, which proves the superiority of E2E decoding.

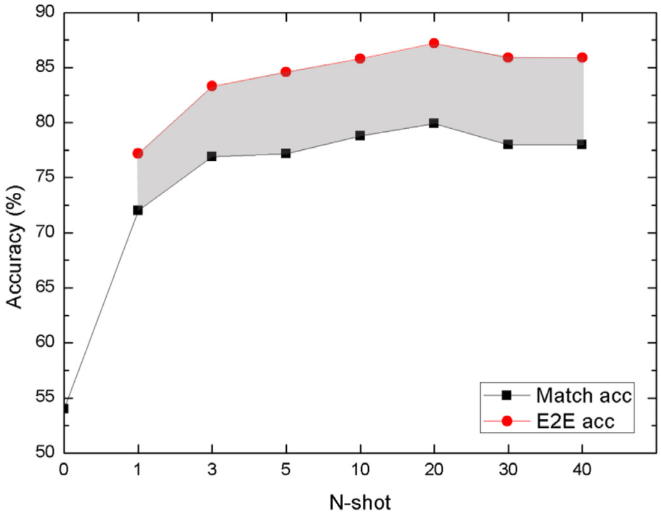


Fig. 6. Experiment results on N-shot recognition of unseen handwritten characters.

Table 2
Experimental results on different scales of training set.

M/total	Testing/total	E2E acc
20%	80%	63.0%
30%	70%	73.1%
40%	60%	80.2%
50%	50%	84.6%
60%	40%	88.5%
70%	30%	90.5%
80%	20%	92.4%

Table 3
Comparison with previous study.

Training class	Testing class	Methods	
		DenseRAN [24]	Our method
500	1000	1.70%	33.6%
1000	1000	8.44%	41.5%
1500	1000	14.71%	63.8%
2000	1000	19.51%	70.6%
2755	1000	30.68%	77.2%

To explore the effect of different M, we fixed N at 10, and changed the ratio of M to total classes(6391) from 20% to 80%. From Table 2, it can be seen that although 80% of character classes in the testing set are unseen in the training set (M/total=20%), our method can still recognize them with an accuracy of 63.0%. Furthermore, the accuracy grows to 84.6% when only 50% classes are used for training.

For better comparison with the previous state-of-the-art approaches [24], we also conduct experiments on 3755 common characters. Following the settings described in [24], we chose 2755 classes in HWDB1.0 and 1.1 as the training set and another 1000 classes in ICDAR-2013 as the testing set. As shown in Table 3, our method exhibits superior performance, which again validates its effectiveness.

4.4. Experiments on recognition of seen characters

In this subsection, we will show the effect of our method on seen characters. We used the offline CASIA-HWDB1.0–1.1 (DB1.1) [27] databases for training, and the test dataset from the 2013 ICDAR Chinese handwriting recognition competition [28] for testing. Both the training set and testing set contains 3755 classes. The experimental result are shown in Table 4, we can see that our method maintains a high performance for seen characters.

Table 4
Experimental results on the recognition of seen characters.

Method	Accuracy
Human Performance [28]	96.13%
DFE-DLQDF [11]	92.72%
HCCR-GoogLeNet [33]	96.26%
DirectMap-CNN-Adaptation [31]	97.37%
M-RBC + IR [25]	97.37%
DenseRAN [24]	96.66%
Our method	96.97%

4.5. Discussion and future work

Advantage Compared with the traditional methods, the RAN addresses OHCCR problem in the perspective of radical learning, and achieves over 90% accuracy on unseen characters given only few support samples. Obviously it is very suitable for OHCCR problem of which the data is expensive, especially when the training classes are incomplete.

Disadvantage Although exceeding the previous methods significantly, the performance of RAN on recognizing unseen classes is far behind human performance, which means that the radical representation learned by RAN is still not effective enough. Besides, the computation of RAN takes more than twice the time of the baseline model.

Future Work In the future, we will find more effective radical representation and try to solve the computing efficiency problem.

5. Conclusion

To achieve the few-shot/zero-shot recognition of OHCCR, in this paper we propose RAN, which introduces a distance metric criterion for radical features to improve its robustness, and integrates an end-to-end decoding strategy with few support samples used for training. Experiments show that our method can effectively recognize unseen Chinese handwritten characters, while maintaining a high performance for seen characters.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgment

This research is supported in part by GD-NSF (no. 2017A030312006), the National Key Research and Development Program of China (no. 2016YFB1001405), NSFC (grant no.: 61673182, 61771199), and GDSTP (Grant No.: 2017A010101027), GZSTP (no. 201704020134).

References

[1] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: International Conference on Learning Representations (ICLR), 2015.

[2] W. Chen, W. Liu, Z. Kira, Y.F. Wang, J. Huang, A closer look at few-shot classification, in: Proceedings of International Conference on Learning Representations (ICLR), 2019.

[3] J. Chung, C. Gulcehre, K.H. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, in: CoRR, 2014. arXiv: 1412.3555.

[4] D.C. Ciresan, U. Meier, Multi-column deep neural networks for offline handwritten chinese character classification, in: Proceedings of International Joint Conference on Neural Networks (IJCNN), 2015, pp. 1–6.

[5] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[6] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[7] T. Jian, J. Liu, Z. Ming, Q. Mei, Visualizing large-scale and high-dimensional data, in: Proceedings of International Conference on World Wide Web (WWW), 2016, pp. 287–297.

[8] S. Lai, L. Jin, W. Yang, Toward high-performance online hccr: a cnn approach with dropdistortion, path signature and spatial stochastic max-pooling, Pattern Recognit. Lett. 89 (2017) 60–66.

[9] C. Li, W. Song, F. Wei, J. Sun, S. Naoi, Beyond human recognition: a cnn-based framework for handwritten character recognition, in: Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 695–699.

[10] C. Liu, M. Nakagawa, Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition, Pattern Recognit. 34 (3) (2001) 601–615.

[11] C. Liu, F. Yin, D. Wang, Q. Wang, Online and offline handwritten Chinese character recognition: benchmarking on new databases, Pattern Recognit. 46 (1) (2013) 155–162.

[12] Y. Liu, L. Jin, S. Lai, Automatic labeling of large amounts of handwritten characters with gate-guided dynamic deep learning, Pattern Recognit. Lett. 119 (2017) 94–102.

[13] T. Long, X. Xu, F. Shen, L. Liu, N. Xie, Y. Yang, Zero-shot learning via discriminative representation extraction, Pattern Recognit. Lett. 109 (2017) 27–34.

[14] L.L. Ma, C.L. Liu, A new radical-based approach to online handwritten Chinese character recognition, in: Proceedings of International Conference on Pattern Recognition (ICPR), 2008, pp. 1–4.

[15] J. Myers, Knowing Chinese character grammar, Cognition 147 (2016) 127–132.

[16] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1988) 696–699.

[17] J. Snell, K. Swersky, R.S. Zemel, Prototypical networks for few-shot learning, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), 2017, pp. 4077–4087.

[18] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), 2014, pp. 3104–3112.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.

[20] D. Tao, L. Liang, L. Jin, Y. Gao, Similar handwritten Chinese character recognition by kernel discriminative locality alignment, Pattern Recognit. Lett. 35 (2014) 186–194.

[21] L. Teng, L. Jin, Building compact mqdf classifier for large character set recognition by subspace distribution sharing, Pattern Recognit. 41 (9) (2008) 2916–2925.

[22] A. Wang, K. Fan, Optical recognition of handwritten Chinese characters by hierarchical radical matching method, Pattern Recognit. 34 (1) (2001) 15–35.

[23] T.Q. Wang, F. Yin, C.L. Liu, Radical-based Chinese character recognition via multi-labeled learning of deep residual networks, in: Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 2017, pp. 579–584.

[24] W. Wang, J. Zhang, J. Du, Z. Wang, Y. Zhu, Denseran for offline handwritten Chinese character recognition, in: Proceedings of International Conference on Frontiers in Handwriting Recognition ICFHR, 2018, pp. 104–109.

[25] Y. Xiao, D. He, Z. Zhou, D. Kifer, C.L. Giles, Improving offline handwritten Chinese character recognition by iterative refinement, in: Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 2017, pp. 5–10.

[26] H. Yang, X. Zhang, F. Yin, C. Liu, Robust classification with convolutional prototype learning, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3474–3482.

[27] F. Yin, Q. Wang, X. Zhang, C. Liu, Icdar 2011 Chinese handwriting recognition competition, in: Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 2011, pp. 1464–1469.

[28] F. Yin, Q.F. Wang, X.Y. Zhang, C.L. Liu, Icdar 2013 Chinese handwriting recognition competition, in: Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 2013, pp. 1464–1470.

[29] M.D. Zeiler, Adadelta: an adaptive learning rate method, CoRR (2012) arXiv:1212.5701.

[30] J. Zhang, Y. Zhu, J. Du, L. Dai, Radical analysis network for zero-shot learning in printed chinese character recognition, in: Proceedings of International Conference on Multimedia and Expo (ICME), 2018, pp. 1–6.

[31] X. Zhang, Y. Bengio, C. Liu, Online and offline handwritten Chinese character recognition: a comprehensive study and new benchmark, Pattern Recognit. 61 (61) (2017) 348–360.

[32] Y. Zhang, S. Liang, S. Nie, W. Liu, S. Peng, Robust offline handwritten character recognition through exploring writer-independent features under the guidance of printed data, Pattern Recognit. Lett. 106 (2018) 20–26.

[33] Z. Zhong, L. Jin, Z. Xie, High performance offline handwritten Chinese character recognition using googlenet and directional feature maps, in: Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 846–850.