2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Radical Counter Network for Robust Chinese Character Recognition

Yunqing Li, Yixing Zhu, Jun Du*, Changjie Wu, Jianshu Zhang
University of Science and Technology of China
National Engineering Laboratory for Speech and Language Information Processing
Hefei, Anhui, P. R. China
lyq123@mail.ustc.edu.cn, zyxsa@mail.ustc.edu.cn, ✉jundu@ustc.edu.cn
wucj@mail.ustc.edu.cn, xysszjs@mail.ustc.edu.cn

***Abstract*—Chinese character recognition has attracted much interest due to its high challenge and various applications. The whole-character modeling method can recognize common characters well but unable to handle unseen situation. Some radical-based modeling methods have successfully achieved great performance in unseen condition but need RNN-based decoder for sequence decoding. Therefore, a compact model which can recognize unseen characters needs to be proposed. First, this paper introduces a novel radical counter network (RCN) to recognize Chinese characters by identifying radicals and spatial structures. The proposed RCN first extracts visual features from input by employing DenseNet as encoder. Then a decoder based on fully connected layer is employed, aiming at synchronously estimating the number of each caption in character. Additionally, we design a multi-task learning to combine global feature extraction capability of whole-character modeling and local feature extraction capability of radical-based modeling, which further improves the model generalization. Experiments on natural scene character dataset demonstrate that the proposed model significantly outperforms WCN by 5.48% and achieve comparable performance with RAN in lower model complexity. That shows great robustness and simplicity of our model.**

Index Terms — Chinese character recognition, radical counter network, multi-task learning, generalization, robustness

I. Introduction

Chinese character recognition remains a challenging work due to its large character categories, high similarity between characters and different application scenarios. Accordingly, some challenging scene character datasets have been proposed [1] [2], which contain distant characters, occluded characters, characters under poor illumination, etc.

To solve this problem, mainly researches can be divided into character-based recognition (CR) and radical-based character recognition (RCR). The CR methods regard a character as a whole (so called the whole-character modeling methods as well), which are efficient and have shown good performance in common Chinese characters recognition [3] [4] [5]. However, it is unable to recognize unseen characters, which is fatal considering the variety of Chinese characters. Moreover, facing with numerous character categories, CR needs numerous outputs accordingly, which increases the indistinguishability of characters further. While as the recently more popular solution, RCR recognizes Chinese characters by analyzing their composition of radicals and structures [6] [7] [8]. Given that more than 20,000 Chinese characters share only about 500 radicals [9], RCR significantly reduces the size of recognition vocabulary. And the ability of decomposing Chinese characters into radicals and structures increases the distinction between similar characters. The most important is that RCR methods have the ability to deal with unseen situations. It seems RCR is a more promising solution.

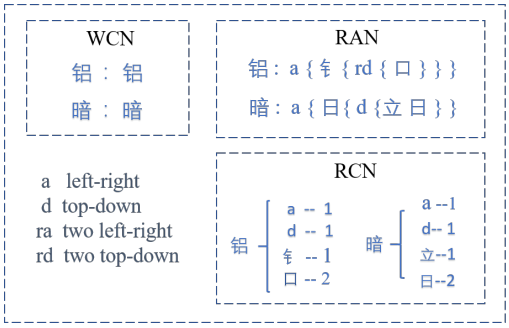

Fig. 1: The comparison of Chinese character representation among WCN, RAN and the proposed model RCN.

As shown in Fig 1, we intuitively compare the representation of character among WCN, RAN and RCN. WCN treats a character as a whole, without regard to its radicals and spatial structures. RAN represents a character with a radical sequence, which follows the rule of tree decomposition. RCN represents the character by the number of its radicals and spatial structures.

Researches on Chinese character recognition has a long history due to its huge practical value. With the development of convolutional neural network (CNN) [10], studies enter a new era. Many networks have got great success on the whole-character recognition, which we collectively call the whole-character network (WCN). [11] proposed a multi-pooling layer on top of CNN for multi-fonts character recognition. [12] captured skeleton features of characters to assist CNN-based classifiers. Recently, to recognize unseen characters, [6] proposed a novel model RAN, which regards a character as a combination of radicals and spatial structures. FewshotRAN [13] further combined deep prototype learning for more robust feature extraction.

However, current mainstream RCR approaches have a com-

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412918

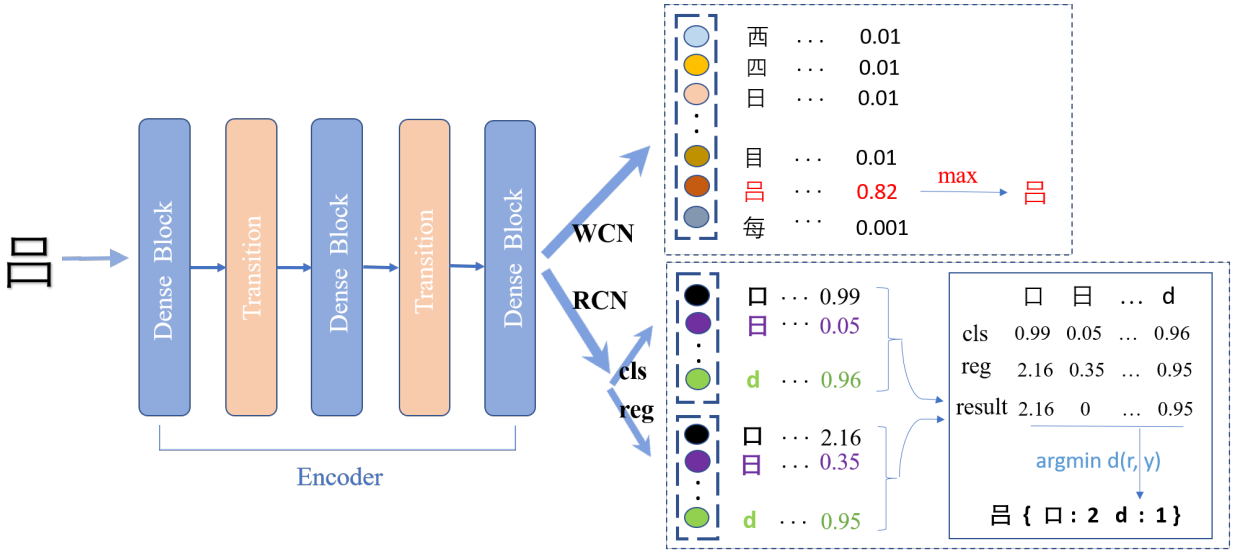

Fig. 2: The comparison of the structure of RCN and WCN. The encoder consists of dense blocks and transition modules. "cls" denotes the classification module. "reg" denotes the regression module. The numbers following the radicals mean the outputs of corresponding neurons. The box denotes the post-preprocess of the outputs."d" denotes top-down structure.

mon shortcoming that they demand a RNN-based decoder, such as RAN [6] and FewshotRAN [13]. Because they represent a character as an ordered sequence of radicals and structures, it is inefficient that the decoder has to predict the sequence one by one. Considering all the benefits and drawbacks of CR and RCR, we propose a novel radical counter network (RCN), thus provide a new solution to Chinese character recognition. RCN has two significant advantages: 1). RCN remains the ability of recognizing unseen characters because the radicals and structures in the inference stage have been learned from the seen characters. 2). RCN is a compact radical-based model which does not need sequentially decoding so that it is as high-efficiency as WCN.

Multi-task learning is one way to achieve inductive transfer between tasks [14] [15]. It has been publicly used in many areas, such as speech recognition [16], computer vision [17], and drug discovery [18]. From the perspective of Chinese character recognition, both global character-level features and local radical-level features are necessary. Treating a character as a whole, WCN demonstrates greater robustness if the input image is occluded. RCN shows advantages when dealing with similar characters for its ability to capture detailed information. Therefore, we design a multi-task model to learn both global and local features and see how two tasks mutually influence.

The main contributions of this study are summarized as follows:

1. We propose a novel model RCN to recognize the unseen and low-frequency characters efficiently.

2. We further design a multi-task network (M-RCN) with dual supervision in radical and character levels, aiming to improve the model generalization.

3. In experiment, M-RCN outperforms WCN with an absolute gain of 5.48% and achieves a comparable result to RAN with lower model complexity in scene Chinese character dataset, which shows great robustness and practicability.

II. NETWORK ACHITECTURE

The structure of RCN contains three parts: a dense encoder, a radical classification module (RCM) and a radical regression module (RRM). The encoder first extracts high-level representations from the input image. Then the RCM judges whether the radicals exist and the RRM estimates the number of existing radicals respectively. Additionally, a multi-task network is designed by sharing the same encoder and combing the outputs of WCN and RCN.

A. RCN

*1) **Training**:* Dense convolutional network (DenseNet) [19] has been proven to be a powerful feature extractor in image classification. So we utilize DenseNet without the final fully connected layer as the encoder to extract high-level features from the input. Given an input image I, through the feature extraction of encoder, we can obtain a three-dimensional array $\boldsymbol{A}$ of size $H \times W \times C$. H, W, C denote orderly the height, width and channels of feature maps. Then we use an adaptive pooling layer $p_{\text{ap}}(\cdot)$ and get a C-dimensional feature vector v.

$$v = p_{\text{ap}}(\boldsymbol{A}) \tag{1}$$

As shown in Fig 2, the decoder of RCN consists of two modules, one is the classification module and the other one is the regression module. We regard spatial structures as special radicals to form the radical vocabulary.

Assuming that there are n elements in the radical vocabulary, RCM contains n neurons. Each neuron outputs the probability o_i^{cls} that each radical exists in the input character.

$$\boldsymbol{o}^{\text{cls}} = \text{sigmoid}\left(f_{\text{cls}}(v)\right) \tag{2}$$

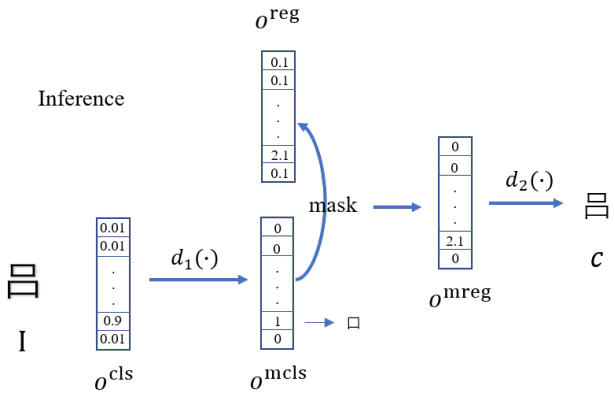

Fig. 3: The illustration of the entire inference process.

$$\boldsymbol{o}^{\text{cls}} = \{o_1^{\text{cls}}, o_2^{\text{cls}}, ..., o_n^{\text{cls}}\} \quad o_i^{\text{cls}} \in [0, 1] \tag{3}$$

$f_{\text{cls}}(\cdot)$ denotes the effect of the fully connected layer in RCM.

Then a threshold function $h(\cdot)$ is set to turn these probabilities into 0 and 1, which are used as masks $\boldsymbol{m} = \{m_1, ..., m_n\} \in \boldsymbol{R}^{1\times n}$. The threshold η is set to 0.5.

$$m_{\text{i}} = h(o_i^{\text{cls}}) = \begin{cases} 0 & o_i^{\text{cls}} < \eta \\ 1 & o_i^{\text{cls}} \geq \eta \end{cases} \tag{4}$$

Considering that a radical may appear in a character more than one time, we utilize RRM to estimate the number of radicals. RRM also consists of n neurons. Each output o_i^{reg} successively corresponds to the same radical as in RCM. Additionally, the classification masks $\boldsymbol{m}$ are multipled on the regression output $\boldsymbol{o}^{\text{reg}}$ to get the predicted radical number $\boldsymbol{o}^{\text{m}} \in \boldsymbol{R}^{1\times n}$:

$$\boldsymbol{o}^{\text{reg}} = f_{\text{reg}}(v) \tag{5}$$

$$\boldsymbol{o}^{\text{m}} = \boldsymbol{m} \cdot \boldsymbol{o}^{\text{reg}} \tag{6}$$

$f_{\text{reg}}(\cdot)$ denotes the effect of the fully connected layer in RRM.

Assuming the ground-truth of o_i^{cls} , o_i^{m} is y_i^{cls} , y_i^{reg} , we use a binary cross entropy for each radical as the classification loss. The distance between o_i^{m} and y_i^{reg} is calculated using squared Euclidean distance function :

$$d(o_i^{\text{m}}, y_i^{\text{reg}}) = ||o_i^{\text{m}} - y_i^{\text{reg}}||_2^2 \tag{7}$$

To balance the two losses, a coefficient λ is set, which is 3 in experiment. The loss function L_{RCN} is definited as:

$$L_{\text{RCN}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=0}^{1} y_{ij}^{\text{cls}} \log(o_{ij}^{\text{cls}}) + \frac{\lambda}{n}\sum_{i=1}^{n} d(o_i^{\text{m}}, y_i^{\text{reg}}) \tag{8}$$

o_{ij}^{cls} denotes the probability that o_i^{cls} belongs to class j. y_{ij}^{cls} denotes the corresponding ground truth.

*2) **Inference**:* The whole process of the inference stage is shown in Fig 3. Given the input image I and the candidate categories Y, we can get the model output $\boldsymbol{o}^{\text{cls}}, \boldsymbol{o}^{\text{reg}}$ and the radical embedding $\boldsymbol{y}^{\text{cls}}, \boldsymbol{y}^{\text{reg}}$ of each category. The final predicted character can be produced as follows:

$$\boldsymbol{o}^{\text{mcls}} = \arg\max_{\boldsymbol{y}^{\text{cls}} \in Y} d_1(\boldsymbol{o}^{\text{cls}}, \boldsymbol{y}^{\text{cls}}) \tag{9}$$

$$\boldsymbol{o}^{\text{mreg}} = \boldsymbol{o}^{\text{mcls}} \cdot \boldsymbol{o}^{\text{reg}} \tag{10}$$

$$c = \arg\min_{\boldsymbol{y}^{\text{reg}} \in Y} d_2(\boldsymbol{o}^{\text{mreg}}, \boldsymbol{y}^{\text{reg}}) \tag{11}$$

c denotes the predicted character. $d_1(\cdot)$ represents the cosine distance metric. $d_2(\cdot)$ represents the Euclidean distance metric. In inference, we choose the category with the closest distance in the candidate set as the prediction result. A character may also be recognized successfully though the radical numbers are not exactly correct, which enables the model better fault tolerance.

B. Multi-task Learning

*1) **Training**:* As shown in Fig 2, WCN utilizes an encoder to extract features and a decoder based on fully connected layers to directly output the predicted character. The proposed RCN has the same encoder but differs in the decoder structure. So we combine these two decoders as a new multi-task network to get both global and local information from the shared encoder. Due to dual guidance in multi-view, both two tasks can mutually promote to improve the generalization of model.

Given the input image I, the output of the WCN

$$\boldsymbol{o}^{\text{WCN}} = \text{softmax}\left(f_{\text{WCN}}(v)\right) \tag{12}$$

$$\boldsymbol{o}^{\text{WCN}} = \{o_1^{\text{WCN}}, o_2^{\text{WCN}} ..., o_K^{\text{WCN}}\} \quad o_i^{\text{WCN}} \in (0, 1) \tag{13}$$

Supposing there are K kinds of characters in sum, the ground-truth of i-th character label is y_i^{WCN}. $f_{\text{WCN}}(\cdot)$ denotes the effect of the fully connected layer in WCN. Then the loss function of multi-task training can be summarized as :

$$L = L_{\text{RCN}} + \mu \sum_{i=1}^{K} y_i^{\text{WCN}} \log(o_i^{\text{WCN}}) \tag{14}$$

μ denotes the balance coefficient between WCN and RCN, which is set to 1e-3 in experiment.

*2) **Inference**:* Since the two tasks share common encoder, which leads to the same result, we only utilize the output parts of RCN to produce the predicted character. The combination of the whole-character output layer only works as a supervision in another view to improve the model generalization. That means there is no additional increase of computation in the inference stage.

III. EXPERIMENT

A. Datasets

We conduct experiments on both printed character dataset and natural scene character dataset to verify the effectiveness of our proposed model.

SCUT-SPCC dataset [11] is a multi-font printed character dataset which contains 280 different fonts. We choose 3,755 commonly used characters with 50 various fonts as the whole

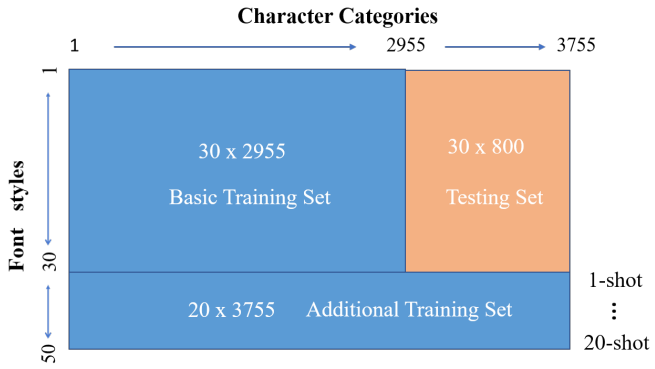

Fig. 4: The illustration of dividing training set and testing set for the printed character experiment.

Fig. 5: Examples with 6 arributes in the CTW dataset.

dataset. These 3,755 characters are composed of 361 radicals and 22 spatial structures. As shown in Fig 4, basic training set contains 2,955 character categories with 30 fonts and the whole testing set consists of the left 800 character categories with the same 30 fonts. Like fewshot learning, N-shot training set is composed of the basic training set and additional 3,755 character categories with other N fonts. On the premise that the characters in the basic training set contain all radicals, the character set division is random. The input images have the size of 48×48.

Chinese Text in the Wild (CTW) [20] is a large dataset of street view images which has approximate 1 million samples. The composition of CTW set is much complicated, mainly containing 6 different attributes. Some examples of CTW dataset are shown in Fig 5. Due to such diversity and complexity, CTW dataset is a challenging set which can truly reflect the practicability of model. The input images are uniformly resized to 32×32 in the following experiments.

B. Implementation Details

The CNN encoder employs DenseNet structure. For different size of training set, we utilize Densenet121 and Densenet169 seperately. Densenet121 mainly consists of three dense blocks, containing (6, 24, 16) bottlenecks successively.

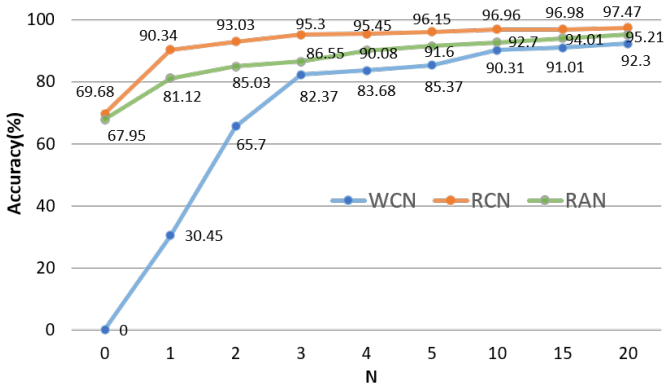

Fig. 6: The performance comparison among WCN, RAN and the proposed RCN with respect to the newly added font styles N.

Each bottleneck is the combination of BatchNorm layer, Relu activate layer, convolutional layer. To avoid overfitting, we add a dropout layer in each bottleneck with the rate of 0.2. A convolutional layer with 64 kernels of size 3×3 is used on top of three dense blocks. Between each two blocks, there is a transition module, which reduces channels with the reduction of 0.5. The growth rate of every bottleneck is set to 32. In Densenet169, there are 4 dense blocks with 3 transition modules. The number of bottlenecks in each dense block is (6, 12, 32, 32). Except for that, other parameters are the same as in Densenet121.

In RCN, the decoder contains a fully connected layer with l neurons. For different datasets, the composition of characters is different. So in experiment on printed character dataset, l is set to 766. And for CTW dataset, l is 830. We utilize Adam optimizer [21] with L2 regularization to avoid overfitting. In multi-task learning, considering there are additional 1,001 character outputs, l is updated to 1,831. Considering the two tasks have different rate of convergence, we set different learning rate accordingly, 5e-4 for the fully connected layer of WCN and 1e-3 for the rest of model.

C. Experiment on Printed Character Dataset

In order to show the great superiority of RCN in zeroshot or fewshot conditions, we first design a few experiments on printed characters.

TABLE I: Comparison of the recognition accuracy rate (%) among WCN, RAN and RCN.

	0-shot	1-shot	2-shot	3-shot	4-shot
WCN	0	30.45	65.7	82.37	83.68
RAN	67.95	81.12	85.03	86.55	90.08
RCN	**69.68**	**90.34**	**93.03**	**95.30**	**95.45**

As comparison, we design the corresponding WCN and RAN whose encoder is consistent with RCN. In experiment, we gradually increase N to see how three modules perform

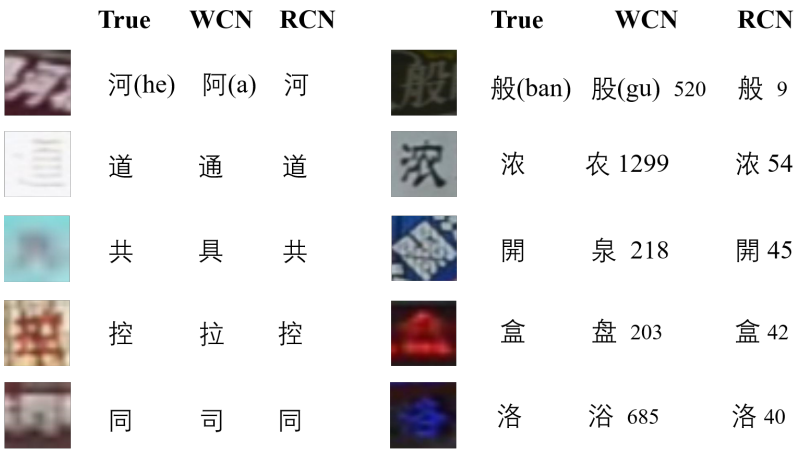

Fig. 7: The comparison examples between WCN and RCN. The samples in left column are similar characters. The samples in right column are low-frequency characters. The numbers following the right characters are their sample numbers in the training set.

respectively. As we can see in Table I, When N is small, RCN significantly outperforms WCN and RAN. In Fig 6, as N increases, RCN can consistently perform best in accuracy. From above, we can conclude that RCN authentically learns radical and spatial structure information from the training set as we expect to recognize the unseen characters. Please note that, without any increase in model complexity, RCN shows great improvement in recognition accuracy.

D. Experiment on Scene Character Dataset

*1) **Comparison with WCN**:* In order to examine the robustness and practicability of RCN, we conduct experiments on CTW dataset. Same as [20], only the top of 1000 character categories are considered for the recognition in the following comparative experiments. Given that there are only three couples of characters which share the same radical composition but differ in spatial structure, we decompose characters in CTW dataset into 415 radicals.

As shown in Table II, M-RCN denotes RCN in multi-task learning. We compare the proposed RCN, M-RCN with DenseNet, which is the corresponding WCN. We also count FLOPs to compare the model complexity. In spite of complex background, low resolution and diverse styles, RCN greatly outperforms WCN by 4.67% with no increase in FLOPs. By jointly training in a typical multi-task network, M-RCN gets further improved by 0.81% with only 0.0034 GFLOPs increase. Compared to the improvement in recognition performance, that is totally ignorable. Please note that, the decoder of WCN is only joinly trained but not participating in testing. That means no additional increase of computaion in inference.

Moreover, we further analyze the specific reasons why RCN can perform better. From Fig 7, we can see that facing with similar characters, WCN easily ignores detailed difference of radicals. For example, character “he” and character “a” only differ in the left-radical, with slanted font WCN recognizes it wrongly. Some similar examples are shown in the left half of Fig 7. Other than that, RCN shows advantages in the low-frequency samples. If certain character rarely emerges in the training set, WCN also has a higher probability to misjudge. As shown in the right half of Fig 7, character “ban” only emerges 9 times in training so that WCN misdeems it for character “gu”, which has 520 training samples. However, different from WCN, RCN lays emphasis on learning radical information, which enables it to recognize the correct character through its radical composition.

TABLE II: Comparison of accuracy rate (%), model GFLOPs among RCN, M-RCN and some other methods.

Method	Accuracy	GFLOPs
AlexNet [2]	73.0	-
Overfeat [2]	76.0	-
ResNet50 [2]	78.2	-
ResNet152 [2]	79.0	-
Google Inception [2]	80.5	-
DenseNet	**79.45**	1.0751
RCN	**84.12**	1.0745
M-RCN	**84.93**	1.0779
RAN [6]	**85.22**	1.0926

To exploit how exactly multi-task training benefits RCN, blocked so that RCN can not recognize any radicals. A part of input character “li” is obscured by the black background so that RCN misses out nearly half radicals. That leads to a wrongly predicted character. However, WCN focuses on the global features so that M-RCN is corrected by dual supervision. Some similar instances are shown in Fig 8. Therefore, there is reason to speculate that multi-task training in our task improves the generalization of original model.

*2) **Comparison with RAN**:* We also compare M-RCN with RAN [6] in recognition accuracy and model complexity. As shown in Table II, the accuracy of M-RCN is only 0.29% lower than RAN, but M-RCN has lower model complexity.

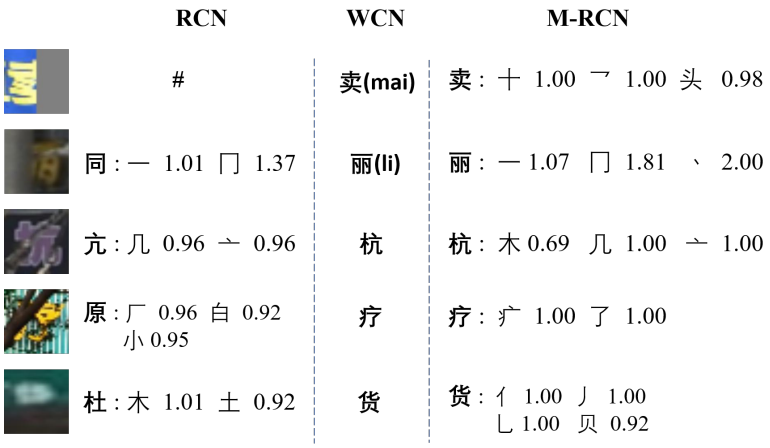

Fig. 8: The comparison examples among WCN, RCN and RCN in multi-task training (M-RCN). The bold character denotes the predicted result of corresponding model, following the predicted radicals and regressing outputs. '#' denotes none.

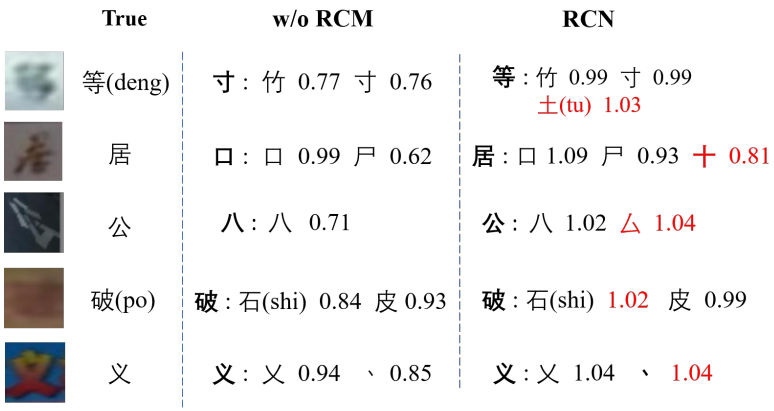

Fig. 9: The comparison examples between RCN w/o RCM and RCN. The bold character denotes the predicted result. The red parts denotes the ones retrieved by RCM. The number following the radical is its regressing output.

As shown in Fig 5, the composition of CTW set is much complicated, even with images of very low definition which are difficult for human to distinguish. Our proposed M-RCN only contains a decoder based on a fully connected layer, with no attention mechanism for radical localization in RAN. Thus the accuracy of M-RCN is comparable to RAN.

RAN treats a character as a sequence of captions and use RNN-based decoder for sequence decoding. So the decoding efficiency is very dependent on sequence length. However, using a fully connected layer as decoder, RCN gets rid of the constraint of sequence length on decoding efficiency. No matter how long the sequence length is, RCN can always decode all captions simultaneously. In the recognition of longer text lines, this feature will make the RCN's efficiency extremely prominent.

E. Ablation Studies

Our innovation is mainly reflected in the design of RCM, RRM and the post-processing of the output. So we conduct a series of ablation experiments on CTW dataset to compare the effectiveness of different modules and key parameters.

*1) **Different Modules**:* The decoder of our proposed RCN consists of RCM and RRM. To further assess the impact of the two modules,we conduct two ablation studies.

TABLE III: Comparison of accuracy rate (%) among RCN w/o RCM, RCN w/o RRM and RCN.

Model	Accuracy	**Acc ↓**
w/o RRM	80.50%	**3.62%**
w/o RCM	78.61%	**4.73%**
RCN	84.12%	-

As shown in III, the accuracy of RCN w/o RRM drops to 80.50%. Because in that way, model can only judge whether the radical exists in the input character, which means the characters with more than two same radicals are definitely wrong. If we remove RCM, the accuracy is significantly reduced by 4.73%. In principle, the cross entropy loss can be easier trained than mean square loss due to the sigmoid function if we just need to judge 0 or 1. So we design the RCM to make judgments in advance. In practice, we show

the obtained contrast samples in Fig 9. Considering that there are 415 radical outputs that we cannot show them all, we only select radicals whose regression output is greater than 0.5. Without RCM, it's easily to miss out one or two radicals for RRM to regress the numbers. For example, character "deng" is composed of three radicals. But RCN w/o RCM misses out the radical "tu" while RCN detects it successfully. Without RCM, although the character may be correctly recognized, the bias that RRM estimates can be greater. For example, character "po" is recognized correctly by RCN w/o RCM, but the regression number of radical "shi" is only 0.84, much more bias comparing to 1.02 regressing by RCN. It seems that the RCM works as a detector to assist the RRM partly.

*2) **Different Distance Metrics**:* In the inference stage, we utilize two distance functions to measure the output of RCM and RRM respectively. We adopt cosine distance and Euclidean distance respectively. The results are shown in IV.

TABLE IV: Comparison of different distance metrics. (Ed denotes the Euclidean distance. Cd denotes the cosine distance.)

d_1	Ed	Cd	Ed	Cd
d_2	Cd	Cd	Ed	Ed
accuracy	83.02%	83.27%	83.87%	**84.12%**

For function $d_1(\cdot)$, the cosine distance is better than the Euclidean distance, and for function $d_2(\cdot)$, the Euclidean distance is better. There may be reasons as follow. Each dimension of RCM output corresponds to a specific primitive, which represents the probability of a radical appearing. The more the cosine distance tends to 1, the more similar the probability distribution. As for the output of RRM, each dimension corresponds to the radical number. Under the premise of correct classification, the number of radicals is more accurately measured by Euclidean distance.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel radical counter network, which is simple, compact yet powerful. In experiment, we show the great superiority of RCN in unseen situations and robust performance in scene character dataset. In the future work, we plan to further investigate the ability of RCN in other language recognition tasks. Moreover, we will continue to expand the model to text line recognition.

V. 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. This work was also funded by Tencent.

REFERENCES

[1] X. Liu, R. Zhang, Y. Zhou, Q. Jiang, Q. Song, N. Li, K. Zhou, L. Wang, D. Wang, M. Liao *et al.*, "Icdar 2019 robust reading challenge on reading chinese text on signboard," *arXiv preprint arXiv:1912.09641*, 2019.

[2] T.-L. Yuan, Z. Zhu, K. Xu, C.-J. Li, T.-J. Mu, and S.-M. Hu, "A large chinese text dataset in the wild," *Journal of Computer Science and Technology*, vol. 34, no. 3, pp. 509–521, 2019.

[3] X.-Y. Zhang, Y. Bengio, and C.-L. Liu, "Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark," *Pattern Recognition*, vol. 61, pp. 348–360, 2017.

[4] J. Du, J.-F. Zhai, J.-S. Hu, B. Zhu, S. Wei, and L.-R. Dai, "Writer adaptive feature extraction based on convolutional neural networks for online handwritten chinese character recognition," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 841–845.

[5] J. Du, J.-S. Hu, B. Zhu, S. Wei, and L.-R. Dai, "A study of designing compact classifiers using deep neural networks for online handwritten chinese character recognition," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 2950–2955.

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

[7] T.-Q. Wang, F. Yin, and C.-L. Liu, "Radical-based chinese character recognition via multi-labeled learning of deep residual networks," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 579–584.

[8] W. Wang, J. Zhang, J. Du, Z.-R. Wang, and Y. Zhu, "Denseran for offline handwritten chinese character recognition," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 104–109.

[9] X. Li and X. Zhang, "The writing order of modern chinese character components," *The journal of modernization of Chinese language education*, 2013.

[10] 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.

[11] Z. Zhong, L. Jin, and Z. Feng, "Multi-font printed chinese character recognition using multi-pooling convolutional neural network," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 96–100.

[12] W. Tang, Y. Su, X. Li, D. Zha, W. Jiang, N. Gao, and J. Xiang, "Cnn-based chinese character recognition with skeleton feature," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 461–472.

[13] T. Wang, Z. Xie, Z. Li, L. Jin, and X. Chen, "Radical aggregation network for few-shot offline handwritten chinese character recognition," *Pattern Recognition Letters*, vol. 125, pp. 821–827, 2019.

[14] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[15] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[16] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8599–8603.

[17] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[18] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, "Massively multitask networks for drug discovery," *arXiv preprint arXiv:1502.02072*, 2015.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[20] T.-L. Yuan, Z. Zhu, K. Xu, C.-J. Li, and S.-M. Hu, "Chinese text in the wild," *arXiv preprint arXiv:1803.00085*, 2018.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.