A Semantic Enhanced Chinese Text Recognition Method Based on Stroke Decomposing

Hang Yu
Beijing University of Posts and Telecommunications
Beijing, China
836671668@qq.com

Xuesong Zhang*
Beijing University of Posts and Telecommunications
Beijing, China
xuesong_zhang@bupt.edu.cn

Zhanchun Gao
Beijing University of Posts and Telecommunications
Beijing, China
gaozc@bupt.edu.cn

2022 IEEE 8th International Conference on Computer and Communications (ICCC) | 978-1-6654-5051-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCC56324.2022.10065939

***Abstract*—Chinese text recognition has been one of the hot topics. Although there are many text recognition methods available, some problems are still not effectively solved, such as the appearance of Chinese characters that do not appear in the training process. Previous Chinese text recognition methods have not thoroughly solved this problem, and many of them are based on English text recognition methods. English words can be split into different character, and Chinese character can be similarly split Chinese characters into multiple stroke sequences, then recognize the stroke sequences, and finally connect the stroke sequences into sentences. Howerver, the same sequence of strokes can form different Chinese characters, so a sequence of strokes can also form different sentences. To solve this confusion problem, a text classification model is trained to rank all possible sentences and then find the one that best matches the normal semantics.**

***Keywords*—OCR,Chinese text recognition,stroke**

I. Introduction

Text recognition has been a hot topic in recent years, and has an important role in many applications. Chinese text recognition is especially important in many scenarios. For example, scene text recognition, document recognition, etc. The current Chinese text recognition methods are based on English text recognition method trained by Chinese datasets. For English text, there are only 26 fixed English characters, while the Chinese characters are close to three thousand characters alone commonly, and the total Chinese characters are about close to one hundred thousand. This will cause the parameters of model to be too large.

Existing Chinese text recognition methods mainly uses generic text recognition models to train on Chinese datasets, which are based on deep learning. These methods fall into two main categories: the first category of methods is not based on language models, they use convolutional neural networks or Transformer to extract visual features to get information about the characters. For example, CRNN [1] uses CNN and RNN to extract visual features and is supervised by the connectionist temporal classification (CTC) loss [2] to maximize the probability of the ground truth, which solves the problem of alignment. But it only uses one dimension feature, which is vulnerable to some natural scenes. In this case, segmentation-based methods are put forward to mitigate this situation. Liao et al. [3] proposes a character attention fully convolutional network which utilizes two dimension features to improve accuracy. The second is based on language models. These methods incorporated semantic prior into visual feature extractor to fully utilize the external language priors. However, those methods all use a single Chinese character as a class, causing the parameters of model too large. There are some methods that study the recognition of individual Chinese characters. Chen et al. [4] splits Chinese character into stroke to address the zero-shot problem, it decomposes a character into a combination of five strokes, including horizontal, vertical, left-falling, right-falling, and turning as shown in Fig. 1. But the recognition of single characters is hardly of practical use, because in practical scenarios, optical character recognition (OCR) usually includes two stages, the first stage is text detection, the output of this stage is the location of text lines. Then the second stage is text recognition, according to the results of text lines returned from text detection, to identify the content of text lines. This structure is currently the most used. The use of the single Chinese character recognition is very limited.

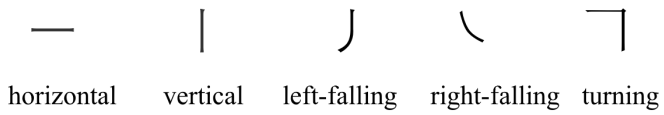

Fig. 1. 5 basic strokes

Inspired by the stroke-level character recognition method [4], we proposed a stroke-based Chinese sentence recognition method. Each character in a sentence is split into a sequence of strokes according to the Chinese strokes. Each character in sentence is decomposed into a combination of five strokes, including horizontal, vertical, left-falling, right-falling, and turning. Chen et al. [4] recognizes individual characters as strokes, and we change the input as well as the labeling method to recognize text lines as sequences of strokes. Thus, the zero-shot problem is not existed anymore. However, the relationship between stroke sequence and Chinese sentences is not one-to-one. Multiple characters may correspond to the same stroke sequence. To solve this problem, we use FastText [5] as pre-trained language model to carry out the text classification

task, obtaining the correct sentence in the test stage. We conduct the experiences on some datasets, including scene dataset, document dataset and web dataset. In summary, our contributions can be listed as follows:

- We applied the stroke-based approach to Chinese text recognition.
- To solve the situation that the stroke sequence does not correspond to the utterance one by one, we added a language module to classify multiple possible utterances to get the correct one.

II. RELATED WORK

A. Language-free Methods

Language-free methods generally use visual features to recognize characters, ignoring the semantic context between charaters, such as CTC-based [1] methods and segmentation-based [3] methods. The CTC-based methods use nonvolutional neural networks (CNN) to extract visual features, and then use recurrent neural network (RNN) to model visual context. Then architecture are trained end-to-end with CTC loss [2] which is mainly used to handle the alignment of input and output labels in the sequence labeling problem. The segmentation-based methods usually segment the characters in a line of text individually and identify the result for each character by means of classification. Such methods usually require character-level annotation, but exsiting datasets are usually in text-level annotation. Liao et al. [3] apply a character attention fully convolutional network to segment characters in pixel-level. It is essentially a classification method, and if undefined categories appear, it will affect the results. Due to the lack of semantic information, the language-free approach is not a good solution for the recognition of low-quality images.

B. Language-based Methods

Language-based methods are usually based on context aware models, which implicitly models language using attention mechanism [6] and Transformer [7]. The attention-based methods follow encoder-decoder architecture, where the encoder extracts of image features images and the decoder generates characters by focusing on relevant features. Lee et al. [6] encodes the input text images horizontally as one-dimensional sequential visual features. The linguistic features are then guided to attend to the corresponding visual features, and then directs the visual features to appear in the corresponding regions with the help of semantic information from the previous time step. ASTER [8] adds a correction module in which spatial transformation modules [9] is employed. MORAN [10] first employs a multi-object correction network to predict the corrected pixel offsets in a weakly supervised manner. The output pixel offsets are further used to generate rectified images, which are further sent to an attention-based decoder for final prediction. SAR [11] is a representative method that uses two-dimensional feature maps for more robust decoding. SEED [12] uses the architecture of ASTER [8] and combine the pre-trained FastText [5] model to guide the visual feature extraction in the training stage, bringing extra linguistic information to the model. ABInet [13] proposes an execution manner of iterative correction for language model which can improve recognition of low quality image. The language model of above method can utilize semantic context to get more accurate results. SRN [14] proposes a novel end-to-end trainable framework of Semantic Reasoning Network to capture the global semantic context through multiple parallel transmission. TransOCR [15] is one of the representative Transformer-based methods, which uses super-resolution prior to guide the training of text recognition.

C. Character-based Methods

Single character recognition is still one of the hot spots for Chinese text recognition. Some methods recognize the entire character, others break down the character into strokes. MCDNN [16] is the first case of using CNN, which assembles 8 models while outperforming humans in recognizing chinese handwritten characters. However, there are many Chinese characters that are similar in shape, and it is difficult for convolutional neural networks to learn visual features. In order to solve the difficulty of distinguishing between similar Chinese characters, Xiao et al. [17] propose the template and instance loss functions for Chinese character recognition. There are also some stroke-based methods. Recently, Chen et al. [4] decomposes character into a strokes sequence. For strokes are the basic components of Chinese characters, the zero-shot Chinese character problem has been solved. Then, they choose the correct character by the similarity of visual features.

III. METHODOLOGY

In this section we describe the proposed method in detail. The method is divided into two stages. The first stage uses the ResNet [18] encoder and Transformer [19] decoder architecture to decompose the text into strokes, and finally obtains the stroke sequence of the sentence. The second stage converts the stroke sequence to text. Since the same stroke sequence may correspond to more than one Chinese sentence, text classification is used here to distinguish the sentences. More specifically, we choose FastText [5] as our text classification model.

A. Text-to-Stroke

The text-to-stroke overall is an encoder-decoder structure as shown in Fig. 2. The encoder is ResNet [18], which extracts the image features. ResNet [18] is now used as the backbone network for extracting image features, which uses residual blocks to solve the problem of network degradation. The input is a three-channel image $I \in H \times W \times 3$. If the image is single-channel, it will expand to three channels. Each image will be resize to 32×100.

Then the image features are input to Transformer decoder [19] for decoding to obtain the corresponding stroke sequence for this text line. Transformer has been widely used in deep learning tasks, the decoder is similar to the original Transformer [19], which includes the masked multi-head attention

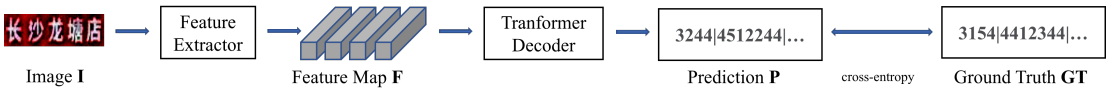

Fig. 2. The architecture of text-stroke network.

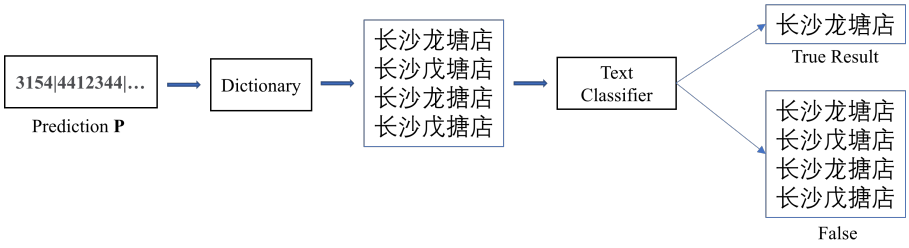

Fig. 3. The architecture of stroke-text network.

module (Masked MHA), the multi-head attention module (MHA), and the feed-forward module. During training, each Chinese character in a sentence needs to be separated by a separator, and a closing symbol needs to be added at the end of the sentence. The loss function is cross-entropy: $l = -\sum_{t=1}^{T} log(g_t)$, where T is the length of stroke sequence , g_t is the ground truth at time t and $p(g_t)$. The maximum of the sequence is set to 200.

B. Stroke-to-Text

The overall architecture of stroke-to-text is shown as Fig. 3. Chen et al. [4] distinguishes the correct Chinese characters by matching visual features. In contrast, in this method, We treat this task as text classification to get the correct sentence. Text classification is the most basic and important task in natural language processing (NLP). We divide the text into two categories, the correct utterances, which usually have normal semantics. The other category is other incorrect statements in the same stroke, which do not have normal semantics. In our method FastText [5] is used to determine the most consistent normal semantic information. FastText is based on the skip-gram model, which is a model in the text classification task. The strokes are arranged into several different utterances, and then FastText [5] is used to classify these utterances. The ones that match the normal semantics and have the highest confidence will be used as the recognition result.

FastText [5] is based on skip-gram [20]. In the skip-gram mechanism, a word is represented by an embedding vector, which is then put to a feed-forward artificial neural network to predict the semantic context. By training the feed-forward network, the embedding vectors are simultaneously optimized with the training of the neural network. After optimization, words with the same semantic meaning have similar embedding vectors. FastText [5] further embeds subwords and uses them to produce the final word embedding vector, which can solve the problem of "out of vocabulary". The model aims to learn a linguistic representation for each word ω. Given a large training corpus represented as a sequence of words $\omega_1, \ldots, \omega_T$, the objective of the skip-gram model is to maximize the following loss function:

$$\sum_{t=1}^{T} \sum_{c \in \mathcal{C}_t} \log p\left(w_c \mid w_t\right)$$

where the context C_t is the set of indices of words surrounding word ω_t.

In the training phase, we arrange and combine the characters corresponding to the strokes according to the strokes sequence to get all possible combinations. We set the label of the correct statement to 0 and the label of the incorrect statement to 1, and then give it to FastText [5] for training. After training, the generated sentences are input into the model and the ones with category 0 are taken as the result. If there are multiple sentences with category 0, the one with the highest confidence is selected as the result.

IV. EXPERIMENTS

In this section, we first present the data set used for the experiments and the basic setup, as well as the evaluation methods. These datasets contains Chinese, English character, and numbers, most of which are in Chinese. This experiment mainly focuses on the splitting of Chinese characters, while English and numbers cannot be split into stroke order. So when we encounter English and numbers, we will not split them. Moreover, we compare the data with existing methods.

A. Datasets

We use the datasets provided by [21] to compare existing text recognition methods.

a) Scene Dataset: Scene dataset includes RCTW [22], ReCTS [23], LSVT [24], ArT [25], and CTW [26], total 636,455 text samples.

b) Web Dataset: Web dataset includes MTWI [27] that contains 20,000 Chinese and English web text images from website.

c) Document Dataset: Synthetic text images about 500,000 samples [21].

B. Implementation Details

We implement our method with PyTorch based on an NVIDIA RTX3090 GPU with 24GB memory. In the text-stroke training stage, each input image is resized to 32×100. The batch size is set to 128. In the stroke-text training stage, when generating the training corpus, the possible utterances are listed, the order is disordered and divided into a training set, a validation set and a test set according to a ratio of 8:1:1.

TABLE I
THE RESULTS OF THE BASELINES ON THREE DATASETS. THE SENTENCE ACCURACY (ACC) IS USED AS THE EVALUATION METRIC. ACC FOLLOW THE PERCENTAGE FORMAT.

Methods	**Datasets**		
	Scene	***Web***	***Document***
CRNN [1]	54.9	56.2	97.4
ASTER [8]	59.4	57.8	97.6
MORAN [10]	54.7	49.6	91.7
SAR [11]	54.9	56.2	97.4
SEED [12]	45.4	31.3	96.1
TransOCR [15]	67.8	62.7	97.9
Ours	59.6	63.4	94.3

C. Results

The results of this experiment are shown in TABLE I. Our method performs very well on the Web dataset, followed by the Scene dataset, and less well on the Document dataset. The Scene dataset and the Document dataset have a higher proportion of common words compared to the web dataset, and TransOCR [15] can achieve good results on these two datasets because the dataset is a scene image, which is basically a common word, and the zero-shot problem has less impact on this dataset. The web dataset is collected on some web pages, including some uncommon words, and the proportion of rare words is higher. The method can achieve better results in the web dataset, which proves that the method is feasible.

V. CONCLUSIONS

We propose a method to decompose utterances into strokes for recognition. The method uses ResNet [18] as well as Transformer to extract visual features, and then transforms the feature vector into a sequence of strokes. Since the stroke sequence and Chinese characters are not one-to-one, which may cause confusion problems, the FastText model [5] is then used to classify all possible utterances to obtain the correct one. The current text line based approach recognizes whole lines of text and relies on the corpus provided in the training phase. If Chinese characters that did not appear in the training phase are encountered in the testing phase, the recognition results are affected. There are also methods based on strokes, but they are single character recognition, and the actual scenario is mainly based on the recognition of text lines. Experiments show that our results have good results in some datasets, proving that the method is practicable.

REFERENCES

[1] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2298–2304, 2017.

[2] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[3] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang, P. Lyu, C. Yao, and X. Bai, "Scene text recognition from two-dimensional perspective," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8714–8721.

[4] J. Chen, B. Li, and X. Xue, "Zero-shot chinese character recognition with stroke-level decomposition," *arXiv preprint arXiv:2106.11613*, 2021.

[5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, no. 5, pp. 135–146, 2017.

[6] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for ocr in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2231–2239.

[7] P. Wang, L. Yang, H. Li, Y. Deng, C. Shen, and Y. Zhang, "A simple and robust convolutional-attention network for irregular text recognition," *arXiv preprint arXiv:1904.01375*, vol. 6, no. 2, p. 1, 2019.

[8] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: An attentional scene text recognizer with flexible rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2035–2048, 2019.

[9] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/33ceb07bf4eeb3da587e268d663aba1a-Paper.pdf

[10] C. Luo, L. Jin, and Z. Sun, "Moran: A multi-object rectified attention network for scene text recognition," *Pattern Recognition*, vol. 90, pp. 109–118, 2019.

[11] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 8610–8617.

[12] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "Seed: Semantics enhanced encoder-decoder framework for scene text recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 528–13 537.

[13] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7098–7107.

[14] D. Yu, X. Li, C. Zhang, T. Liu, J. Han, J. Liu, and E. Ding, "Towards accurate scene text recognition with semantic reasoning networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 113–12 122.

[15] J. Chen, B. Li, and X. Xue, "Scene text telescope: Text-focused scene image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 026–12 035.

[16] D. Cireşan and U. Meier, "Multi-column deep neural networks for offline handwritten chinese character classification," in *2015 international joint conference on neural networks (IJCNN)*. IEEE, 2015, pp. 1–6.

[17] Y. Xiao, D. Meng, C. Lu, and C.-K. Tang, "Template-instance loss for offline handwritten chinese character recognition," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 315–322.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their composi-

tionality," *Advances in neural information processing systems*, vol. 26, 2013.

[21] J. Chen, H. Yu, J. Ma, M. Guan, X. Xu, X. Wang, S. Qu, B. Li, and X. Xue, "Benchmarking chinese text recognition: Datasets, baselines, and an empirical study," *arXiv preprint arXiv:2112.15093*, 2021.

[22] B. Shi, C. Yao, M. Liao, M. Yang, P. Xu, L. Cui, S. Belongie, S. Lu, and X. Bai, "Icdar2017 competition on reading chinese text in the wild (rctw-17)," in *2017 14th iapr international conference on document analysis and recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1429–1434.

[23] R. Zhang, Y. Zhou, Q. Jiang, Q. Song, N. Li, K. Zhou, L. Wang, D. Wang, M. Liao, M. Yang *et al.*, "Icdar 2019 robust reading challenge on reading chinese text on signboard," in *2019 international conference on document analysis and recognition (ICDAR)*. IEEE, 2019, pp. 1577–1581.

[24] Y. Sun, Z. Ni, C.-K. Chng, Y. Liu, C. Luo, C. C. Ng, J. Han, E. Ding, J. Liu, D. Karatzas *et al.*, "Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1557–1562.

[25] C. K. Chng, Y. Liu, Y. Sun, C. C. Ng, C. Luo, Z. Ni, C. Fang, S. Zhang, J. Han, E. Ding *et al.*, "Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1571–1576.

[26] 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.

[27] M. He, Y. Liu, Z. Yang, S. Zhang, C. Luo, F. Gao, Q. Zheng, Y. Wang, X. Zhang, and L. Jin, "Icpr2018 contest on robust reading for multi-type web images," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 7–12.