Chinese Text Recognition with A Pre-Trained CLIP-Like Model Through Image-IDS Aligning

Haiyang Yu, Xiaocong Wang, Bin Li[*], Xiangyang Xue
Shanghai Key Laboratory of Intelligent Information Processing
School of Computer Science, Fudan University
{hyyu20, xcwang20, libin, xyxue}@fudan.edu.cn

Abstract

Scene text recognition has been studied for decades due to its broad applications. However, despite Chinese characters possessing different characteristics from Latin characters, such as complex inner structures and large categories, few methods have been proposed for Chinese Text Recognition (CTR). Particularly, the characteristic of large categories poses challenges in dealing with zero-shot and few-shot Chinese characters. In this paper, inspired by the way humans recognize Chinese texts, we propose a two-stage framework for CTR. Firstly, we pre-train a CLIP-like model through aligning printed character images and Ideographic Description Sequences (IDS). This pre-training stage simulates humans recognizing Chinese characters and obtains the canonical representation of each character. Subsequently, the learned representations are employed to supervise the CTR model, such that traditional single-character recognition can be improved to text-line recognition through image-IDS matching. To evaluate the effectiveness of the proposed method, we conduct extensive experiments on both Chinese character recognition (CCR) and CTR. The experimental results demonstrate that the proposed method performs best in CCR and outperforms previous methods in most scenarios of the CTR benchmark. It is worth noting that the proposed method can recognize zero-shot Chinese characters in text images without fine-tuning, whereas previous methods require fine-tuning when new classes appear. The code is available at https://github.com/FudanVI/FudanOCR/tree/main/image-ids-CTR.

1. Introduction

In recent decades, most researchers have focused on exploring Chinese character recognition (CCR) [13, 25, 40, 37, 41], few methods are dedicated to tackle Chinese Text Recognition (CTR). Unlike Latin characters, Chinese characters have a large number of categories and complex internal structures, which lead to zero-shot (*i.e.*, characters in test sets are unseen in training sets) and few-shot problems in practical applications. The conventional framework for CTR should be fine-tuned with the updated alphabet when a new Chinese character appears. However, humans are able to easily match unseen character images with the corresponding characters in their stsndard (e.g. printed) forms. Thus, the question is – *Can a model recognize Chinese texts like humans*?

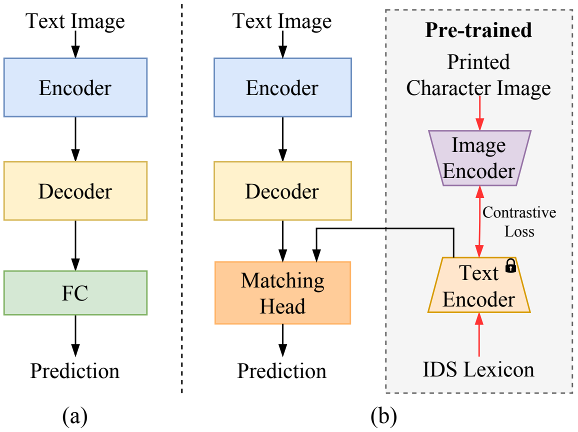


Figure 1. Comparison between the framework of previous methods (a) and that of the proposed method (b). The data flow of the pre-training stage is in red.

To tackle the zero-shot problem, existing CCR methods rely on predicting radical or stroke sequences to recognize characters. For example, some radical-based methods [32, 31] are proposed to decompose Chinese characters at the radical level and predict corresponding radical sequences to determine final predicted characters. Recently, a stroke-based method [5] has been proposed to decompose Chinese characters into stroke sequences, offering a fundamental solution to the zero-shot problem in CCR. These methods are based on relatively complex networks so that

[*]Corresponding Author

they are not suitable for adoption in CTR models to solve zero-shot and few-shot problems. In addition, most scene text recognition models [15, 21] adopt an encoder-decoder framework, which utilizes a fully connected layer to classify characters (as shown in Figure 1(a)). However, these methods require to be fine-tuned when a new character appears, which is inconvenient in practical applications. Furthermore, these methods fail to account for the aforementioned unique characteristic of Chinese characters.

For native Chinese speakers, their initial learning is to recognize individual Chinese characters. In this stage, they also learn how to decompose each Chinese character into the corresponding radical sequence. When reading a text line, they first locate the position of each character and then compare it with the standard characters they have learned to determine its category. For unseen characters, people can use their knowledge of radicals and structures to deduce their categories.

Inspired by the way humans recognize Chinese texts, we propose a two-stage framework (as shown in Figure 1(b)) to address the challenge of CTR. The proposed framework consists of a CCR-CLIP pre-training stage and a CTR stage. In the first stage, we introduce a CLIP-like model, named CCR-CLIP, to learn the canonical representations of Chinese characters through aligning printed character images and their corresponding Ideographic Description Sequences (*i.e.*, radical sequences) in an embedding space. Similar to CLIP [24], the CCR-CLIP model comprises an image encoder and a text encoder, and is trained with a contrastive loss between embeddings of character images and embeddings of radical sequences. To ensure that the image encoder extracts features that are independent of font styles, we also introduce a contrastive loss between input images having the same label in a training batch. After pre-training, the text encoder can output the canonical representations of given radical sequences. In the CTR stage, the learned canonical representations are employed to supervise the CTR model, which is a conventional encoder-decoder framework without a fully connected layer after the decoder. During inference, the model predicts each character in a text image by calculating the similarity between the learned canonical representations and the extracted character embedding. Thus, it is able to recognize zero-shot Chinese characters without fine-tuning. We conduct extensive experiments to validate the effectiveness of the proposed method. Specifically, we train the CCR-CLIP model on several Chinese character recognition benchmarks to evaluate its performance on CCR. The experimental results show that the CCR-CLIP model can robustly recognize Chinese characters in zero-shot settings. Furthermore, our experiments on a CTR benchmark demonstrate that the proposed method outperforms previous methods in most cases.

In summary, our contributions are as follows:

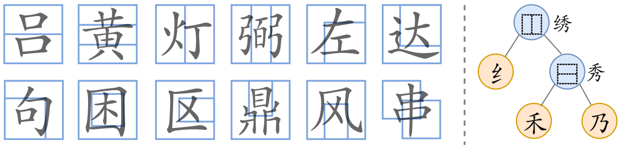


Figure 2. Twelve basic structures represented in blue lines (left) and an example of decomposition at the radical level (right).

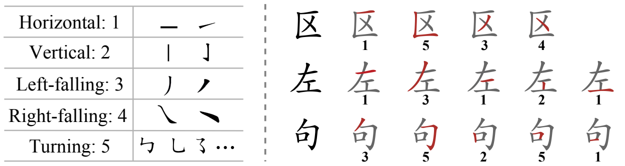


Figure 3. Five categories of strokes for Chinese characters (left) and some examples of decomposition at the stroke level (right).

- Drawing inspiration from how humans recognize Chinese texts, we propose a two-stage framework for CTR, which comprises a CCR-CLIP pre-training stage and a CTR stage.
- We adopt the CLIP architecture to establish a CCR-CLIP pre-trained model to learn the canonical representations of Chinese characters.
- Benefiting from the learned canonical representations, the proposed method can recognize zero-shot characters in Chinese text images without fine-tuning.
- Extensive experiments validate that the CCR-CLIP model outperforms previous CCR methods by a clear margin. Furthermore, the proposed two-stage framework for CTR achieves better performance than previous methods, particularly when training data is scarce.

2. Preliminaries

2.1. Background Knowledge of Chinese Characters

According to Chinese national standard GB18030-2005[1], there are 70,244 classes of Chinese characters, 3,755 of which are commonly-used Level-1 characters. Although Chinese characters have complex inner structures, each Chinese character can be decomposed into the corresponding radical or stroke sequence in a specific order.

Radicals. As shown in Figure 2(right), each Chinese character can be represented as a radical tree, which can be transformed into the corresponding Ideographic Description Sequence (IDS). IDS is defined by Unicode and is composed of radicals and basic structures. Specifically, there are 514 radicals and twelve basic structures (see Figure 2(left)) for 3,755 commonly-used Level-1 characters.

[1] https://zh.wikipedia.org/wiki/GB_18030

Strokes. There are five categories of strokes for Chinese characters according to Chinese national standard GB18030-2005. As shown in Figure 3(left), each category of stroke may contain several instances. Some examples of decomposing Chinese characters at the stroke level are shown in Figure 3(right).

2.2. Related Work

Chinese Character Recognition (CCR). Early Chinese character recognition (CCR) methods typically rely on hand-crafted features [14, 28, 3]. With the development of deep learning, CNN-based methods like MCDNN [7] have achieved remarkable success in extracting robust features of Chinese characters, approaching human-level performance on handwritten CCR tasks in the ICDAR 2013 competition [36]. To address the zero-shot problem in CCR, some methods [22, 16, 1, 18] have been proposed to predict the radical sequences of input character images. For instance, Wang *et al.* [32] used a DenseNet-based encoder [12] to extract character features and an attention-based decoder to predict the corresponding radical sequence. Although such radical-based methods can partially alleviate the zero-shot problem, predicting radical sequences is more time-consuming than character-based methods. Recently, some methods attempt to decompose Chinese characters into stroke sequences to address the zero-shot problem. For example, SD [5] decomposes each Chinese character into a sequence of strokes and employs a feature-matching strategy to address the one-to-many problem between a Chinese character and multiple stroke sequences. Although these CCR methods achieve satisfying performance on various CCR datasets, their complex structures make them unsuitable for the CTR task.

Chinese Text Recognition (CTR). Scene text recognition has made significant strides in recent years. Early CTC-based text recognition methods [30, 26, 9] tend to combine CNN and RNN to extract image features and be optimized through the CTC loss [10]. To address the issue of curved texts, some methods such as ASTER [27] and MORAN [21] have been proposed to transform curved text images into horizontal ones. These methods have achieved promising results in curved text recognition. To incorporate semantic information into recognition models, some methods such as SEED [23] and ABINet [8] introduce an additional language module. Despite the impressive performance of existing methods on Latin text recognition benchmarks, CTR remains a challenging task [6]. To address this problem, a recent work [6] focuses on developing a CTR benchmark and evaluating the performance of mainstream text recognition methods. In addition, the authors proposed to introduce the radical-level supervision to improve the performance of baseline models on the CTR benchmark. However, there are still two unsolved problems: 1) These methods struggle with zero-shot and few-shot problems, which are inevitable in practical applications. 2) When a new character is supplemented in the alphabet, these models should be fine-tuned with the updated alphabet.

3. Methodology

In this paper, we present a novel two-stage framework for Chinese text recognition. The proposed method consists of two stages: the CCR-CLIP pre-training stage and the CTR stage. In the pre-training stage, we develop a CLIP-like model, called CCR-CLIP, which is adopted to learn canonical representations of Chinese characters. The learned representations serve as a guidance for the following CTR model. The architecture of the proposed method is depicted in Figure 4. Next, we provide a detailed introduction to each stage in the proposed method.

3.1. CCR-CLIP Pre-training Stage

Similar to CLIP [24], the proposed CCR-CLIP model consists of an image encoder and a text encoder. The image encoder is responsible for extracting the visual features of the input character image, while the text encoder extracts the features of the corresponding radical sequence. Finally, two contrastive losses are utilized to supervise this model. We train the CCR-CLIP model using printed Chinese character images, and the pre-trained text encoder is used to generate canonical representations for all candidate Chinese characters.

Image Encoder. ResNet [11] is a widely adopted feature extractor and plays a crucial role in optical character recognition tasks [35, 31]. In the CCR-CLIP model, we use ResNet-50 to extract the feature maps $\mathbf{F}^c \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C}$ from an input printed Chinese character image. To represent the input image with a 1-D vector, we employ the global average pooling [17] to compress the feature maps $\mathbf{F}^c$:

$$\mathbf{f}^c = \text{GlobalAvgPool}(\mathbf{F}^c) \tag{1}$$

where $\mathbf{f}^c \in \mathbb{R}^{1\times C}$ denotes the compressed feature vector. At last, we project $\mathbf{f}^c$ into the embedded visual-feature space:

$$\mathbf{I} = \mathbf{f}^c\mathbf{W}^c \tag{2}$$

where $\mathbf{I}$ represents the embedded visual features of the input Chinese character image, $\mathbf{W}^c \in \mathbb{R}^{C\times C'}$ denotes the projection matrix, and C' is the dimensionality for alignment.

Text Encoder. In this paper, we regard the corresponding radical sequence $\mathbf{R} = \{r_1, r_2, ..., r_l\}$ as the caption of the input Chinese character image, where l denotes the length of the radical sequence and r_l is an "END" token. The text encoder consists of K layers of Transformer encoder [29] and an embedding layer. Through the Transformer encoder, $\mathbf{R}$ is encoded into $\mathbf{F}^r = \{\mathbf{f}^r_1, \mathbf{f}^r_2, ..., \mathbf{f}^r_l\}$,

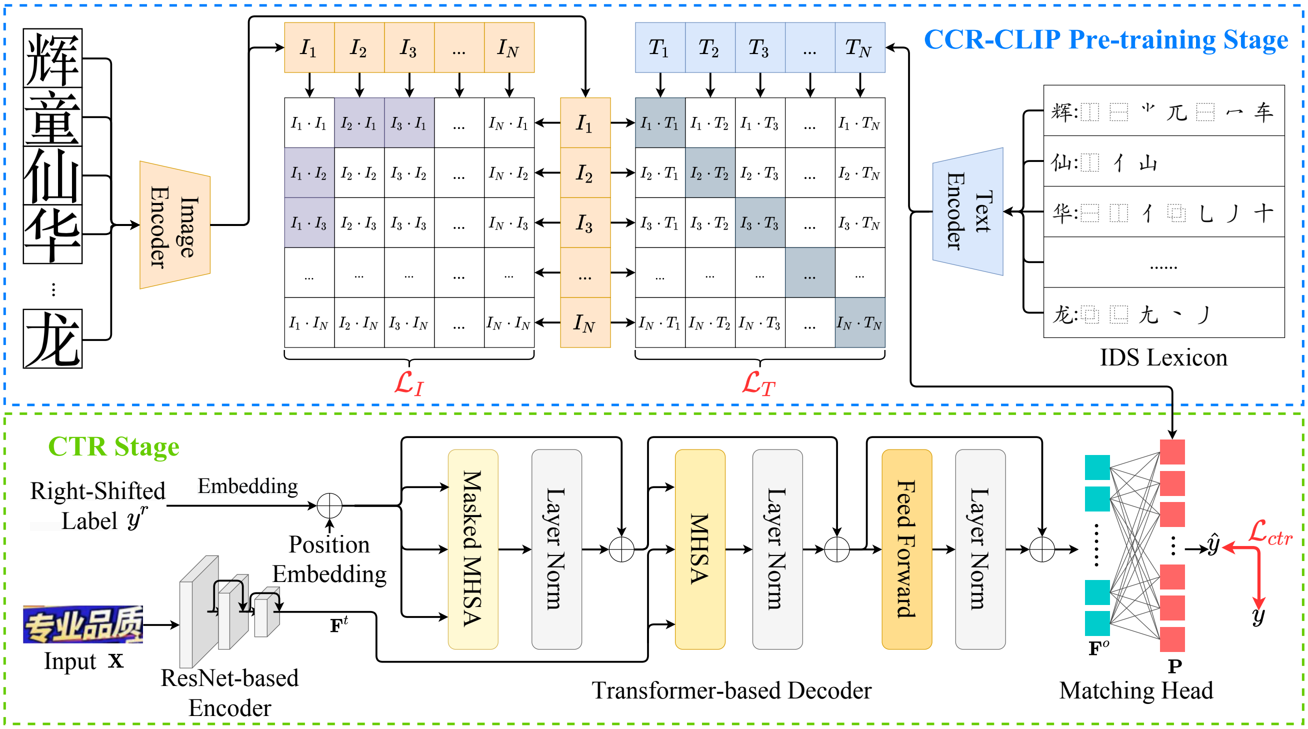


Figure 4. Overall architecture of the proposed method, consisting of a CCR-CLIP pre-training stage and a CTR stage. After being pre-trained at the pre-training stage, the CCR-CLIP model produces canonical representations of Chinese characters for the CTR model. 'MHSA' represents the multi-head self-attention mechanism.

where $\mathbf{f}_l^r \in \mathbb{R}^{1\times D}$ is regarded as the whole features of $\mathbf{R}$. Similar to the image encoder, we project $\mathbf{f}_l^r$ into $\mathbf{T}$:

$$\mathbf{T} = \mathbf{f}_l^r \mathbf{W}^r \tag{3}$$

where $\mathbf{W}^r \in \mathbb{R}^{D\times C'}$ denotes the projection matrix.

Loss Function. We employ a contrastive loss $\mathcal{L}_T$ to align the extracted visual features of a Chinese character image and the features of its corresponding radical sequence. For a training batch with N character samples, the loss function $\mathcal{L}_T$ is calculated as follows:

$$\begin{aligned}\mathcal{L}_T = &-\sum_{j=1}^{N} \log \frac{\exp(\mathbf{I}_j \cdot \mathbf{T}_j)}{\sum_{n=1}^{N} \exp(\mathbf{I}_j \cdot \mathbf{T}_n)} \\ &-\sum_{j=1}^{N} \log \frac{\exp(\mathbf{I}_j \cdot \mathbf{T}_j)}{\sum_{n=1}^{N} \exp(\mathbf{I}_n \cdot \mathbf{T}_j)}\end{aligned} \tag{4}$$

where $\mathbf{I}_j$ and $\mathbf{T}_j$ represent the embedded visual features and radical sequence features of the j-th sample in a data batch, respectively.

To reduce the prediction errors caused by various font styles and similar characters, we additionally introduce a contrastive loss $\mathcal{L}_I$ between the visual features of input images having the same label in the batch. Given a data batch $\mathcal{B} = \{(\mathbf{C}_1, \mathbf{R}_1), (\mathbf{C}_2, \mathbf{R}_2), ..., (\mathbf{C}_N, \mathbf{R}_N)\}$, $\mathbf{C}_i$ and $\mathbf{R}_i$ represent the i-th Chinese character image and its corresponding radical sequence, respectively. Through the image encoder, the i-th character image $\mathbf{C}_i$ is encoded into the corresponding visual features $\mathbf{I}_i$. Thus, the loss function $\mathcal{L}_I$ is computed as follows:

$$\mathcal{L}_I = -\sum_{j=1}^{N} \log \frac{\sum\limits_{\mathbf{I}' \in \mathcal{U}_j} \exp(\mathbf{I}_j \cdot \mathbf{I}')}{\sum_{n=1}^{N} \exp(\mathbf{I}_j \cdot \mathbf{I}_n)} \tag{5}$$

where $\mathcal{U}_j$ represents the set of visual features that have the same corresponding radical sequence $\mathbf{R}_j$. Finally, the overall loss function of the CCR-CLIP model is as follows:

$$\mathcal{L}_{pre} = \mathcal{L}_T + \lambda \mathcal{L}_I \tag{6}$$

where λ is the trade-off coefficient for balancing the two loss items. The experimental results of selecting λ are shown in the Supplementary Material.

3.2. CTR Stage

Taking radical sequences of all candidate characters as input, the pre-trained text encoder can produce their canonical representations $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2 ..., \mathbf{p}_K]$, which are utilized as the supervision at the CTR stage. $\mathbf{p}_k$ denotes the canonical representation of the k-th candidate character and K is the number of candidate characters. For the CTR model, we adopt a conventional encoder-decoder framework that consists of a ResNet-based encoder, a Transformer-based decoder, and a matching head.

ResNet-based Encoder. Given the input text image $\mathbf{X}$, the ResNet-based encoder is employed to extract its visual features $\mathbf{F}^t$. We modify some layers in the original ResNet-34. First, we replace the 7×7 kernel of the first convolution layer with a 3×3 kernel since the smaller kernel can capture more details for recognizing text images. Additionally, we remove the last convolution block to reduce the number of parameters in the encoder, thereby improving the efficiency of feature extraction. Finally, we remove the max pooling layer of the third convolution block in ResNet-34 to reserve more visual features for the subsequent decoder.

Transformer-based Decoder. As shown in Figure 4, the Transformer-based decoder consists of three modules: the masked multi-head self-attention (MHSA) module, the MHSA module, and the feed-forward module. The masked MHSA module takes the right-shifted ground truth y^r as input and captures the semantic dependence between characters. The MHSA module calculates the attention weights between the extracted visual features $\mathbf{F}^t$ and y^r. Finally, the weighted features are fed into the feed-forward module to extract deeper features $\mathbf{F}^o \in \mathbb{R}^{T \times C}$, where T indicates the length of the text, and $\mathbf{F}^o_i \in \mathbb{R}^{1 \times C}$ represents the feature of the i-th character in the input text image.

Matching Head. The previous methods [27, 20] simply utilize a prediction head, *i.e.*, a fully connected layer, to generate the final prediction $\hat{y} = \mathrm{Softmax}(\mathbf{W}^t\mathbf{F}^o + \mathbf{b})$, where $\mathbf{W}^t$ and $\mathbf{b}$ represent the linear transformation and the bias of the prediction head, respectively. Different from these methods, we use the canonical representations of candidate characters $\mathbf{P}$ to match the features of input text image $\mathbf{F}^o$. Thus, the final prediction are generated by:

$$\hat{y} = \mathrm{Softmax}(\mathbf{P}\mathbf{F}^o) \tag{7}$$

Loss Function. The learning objective supervising the CTR model contains two terms:

$$\mathcal{L}_{ctr} = \sum_{\mathbf{f} \in \mathbf{F}^o} (-\log p(y|\mathbf{f}) + \beta R(\mathbf{p}_y, \mathbf{f})) \tag{8}$$

where $-\log p(y|\mathbf{f})$ is the cross-entropy loss, $R(\mathbf{p}_y, \mathbf{f})$ represents the regularization term, and β is a hyperparameter to balance these two terms. The experiment for choosing β is shown in the Supplementary Material. $p(y|\mathbf{f})$ is calculated by:

$$p(y|\mathbf{f}) = \frac{\exp(\mathbf{p}_y \cdot \mathbf{f})}{\sum_{\mathbf{p}_i \in \mathbf{P}} \exp(\mathbf{p}_i \cdot \mathbf{f})} \tag{9}$$

As shown in [34], a regularization term is introduced to avoid overfitting on seen classes, which is defined as:

$$R(\mathbf{p}_y, \mathbf{f}) = ||\mathbf{p}_y - \mathbf{f}||_2^2 \tag{10}$$

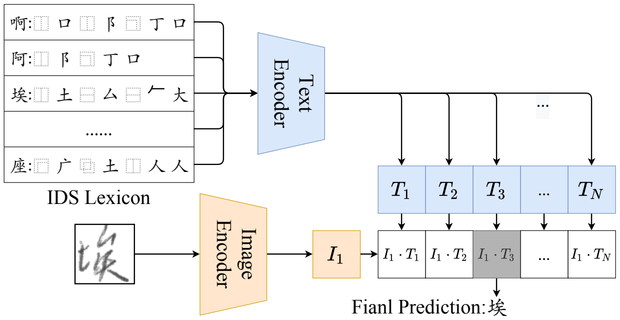


Figure 5. The test process of the CCR-CLIP model for Chinese character recognition.

4. Experiments

Datasets. Extensive experiments on both CCR and CTR are conducted to validate the effectiveness of the proposed method. The adopted datasets are introduced in the following. Examples of each dataset are shown in the Supplementary Material.

- **HWDB1.0-1.1** [19] contains 2,678,424 handwritten Chinese character images with 3,881 classes. This dataset is collected from 720 writers and covers 3,755 commonly-used Level-1 Chinese characters.

- **ICDAR2013** [36] contains 224,419 handwritten Chinese character images with 3,755 classes, which are collected from 60 writers.

- **CTW** [38] is collected from street views, containing 812,872 Chinese character images with 3,650 classes, where 760,107 character images are used for training and 52,765 images are used for testing. This dataset is more challenging due to its complex backgrounds and various fonts.

- **CTR Benchmark** [6] collects four types of Chinese text recognition datasets including scene, web, document, and handwriting. Training, validation and test datasets are divided for each type. In this paper, to fully explore the performance on Chinese texts, we filter out those samples containing non-Chinese characters. Details about these four adopted datasets are introduced in the Supplementary Material.

Evaluation Metrics. Following the previous CCR works [32, 2, 39, 33], we select Character ACCuracy (CACC) as the evaluation metric for CCR. We follow [6] to adopt two mainstream metrics to evaluate our method in CTR: Line ACCuracy (LACC) and Normalized Edit Distance (NED). LACC is defined as:

$$\mathrm{LACC} = \frac{1}{S}\sum_{i=1}^{S} \mathbb{I}(\hat{\mathbf{y}}_i = \mathbf{y}_i) \tag{11}$$

HWDB	m for character Zero-Shot Setting				
	500	1000	1500	2000	2755
DenseRAN [32]	1.70%	8.44%	14.71%	19.51%	30.68%
HDE [2]	4.90%	12.77%	19.25%	25.13%	33.49%
SD [5]	5.60%	13.85%	22.88%	25.73%	37.91%
CUE [22]	7.43%	15.75%	24.01%	27.04%	40.55%
Ours	**21.79%**	**42.99%**	**55.86%**	**62.99%**	**72.98%**
DMN [16]	66.33%	79.09%	84.14%	86.79%	88.98%
CMPL [1]	72.49%	80.57%	84.40%	86.47%	89.29%
CCD [18]	90.93%	94.10%	94.58%	95.55%	-
Ours	**93.80%**	**94.97%**	**95.35%**	**95.71%**	**95.73%**

Table 1. Results in character zero-shot settings. m represents that samples of the first m classes are used for training in CCR zero-shot settings. The results in the top row are only based on HWDB while the results in the bottom row are obtained with additional printed character images during training.

Method	ICDAR2013	CTW	AIT (ms)
ResNet [11]	96.83%	79.46%	**12**
DenseNet [12]	95.90%	79.88%	89
DenseRAN [32]	96.66%	85.56%	1666
FewshotRAN [31]	96.97%	86.78%	83
Template+Instance[33]*	*97.45%*	-	-
RAN [39]	93.79%	81.80%	117
HDE [2]	97.14%	**89.25%**	29
SD [5]	96.28%	85.29%	567
Ours	**97.18%**	85.78%	14

Table 2. Comparison of performance and average inference time (AIT). Methods marked with '*' use additional template character images at the training stage.

where S is the number of text images; $\mathbb{I}$ denotes the indicator function; $\hat{\mathbf{y}}_i$ and $\mathbf{y}_i$ denote the prediction and the label of the i-th text image, respectively. NED is defined as:

$$\text{NED} = 1 - \frac{1}{S}\sum_{i=1}^{S}\text{ED}(\hat{\mathbf{y}}_i, \mathbf{y}_i)/\text{Maxlen}(\hat{\mathbf{y}}_i, \mathbf{y}_i) \tag{12}$$

where "ED" and "Maxlen" denote the edit distance and the maximum sequence length, respectively.

Implementation Details. Our method is implemented with PyTorch, and all experiments are conducted on an NVIDIA RTX 4090 GPU with 24GB memory. The Adam optimizer is adopted to train the model with an initial learning rate 10^{-4}, and the momentums β_1 and β_2 are set to 0.9 and 0.98, respectively. The batch size is set to 128. For fair comparison with previous methods, the input sizes for CCR and CTR are 32 × 32 and 32 × 256, respectively. In the text encoder, the number of Transformer encoder layers is empirically set to 12.

4.1. Results on Chinese Character Recognition

Although the primary objective of the CCR-CLIP model is to generate canonical representations of Chinese characters through aligning printed character images and IDS, it also has the potential to be adapted to recognize Chinese character images through image-IDS matching. The process of inference is depicted in Figure 5.

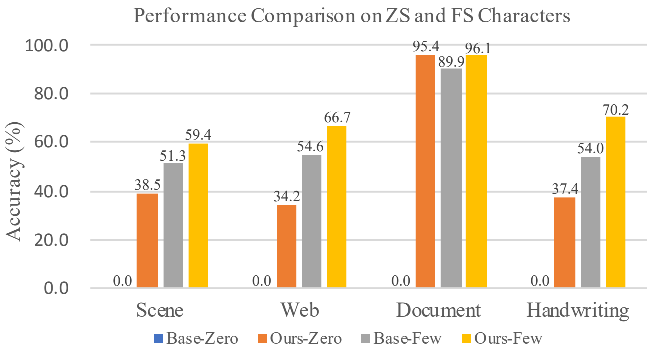


Figure 6. Performance comparison between the baseline model TransOCR and the proposed method in recognizing the zero-shot (ZS) and few-shot (FS) Chinese characters.

Experiments in Zero-shot Settings. Due to the significantly larger alphabet size of Chinese characters, the zero-shot problem is inevitable in practical applications. To address this problem, we follow the approach of [5] and construct corresponding datasets for character zero-shot settings. Specifically, we collect samples with labels falling in the first m classes to form the training set, and collect those in the last k classes for testing. For the handwritten character dataset HWDB, m ranges in {500, 1000, 1500, 2000, 2755}, and k is set to 1000.

The experimental results reported in Table 1 are grouped according to whether printed character images are utilized during the training stage. In the setting of no printed character images used for training, the CCR-CLIP model achieves an improvement of 28.37% in average for the character zero-shot settings, compared with CUE [22]. These results demonstrate the effectiveness of the proposed method. Furthermore, we also incorporate additional printed character images during training, following the approach of [16]. The experimental results show that the proposed CCR-CLIP model still outperforms the compared methods in all character zero-shot settings. The additional experimental results on other datasets and radical zero-shot settings are reported in the Supplementary Material.

Experiments in Non-zero-shot Settings. In contrast to zero-shot settings, we train the proposed CCR-CLIP model using all training samples in non-zero-shot settings, where all characters in the test dataset are covered by the training dataset. For handwritten characters, we use HWDB1.0-1.1 as the training set and ICDAR2013 as the test set. The experimental results reported in Table 2 show that the proposed method achieves the second-best performance, trailing only the template-instance method [33] that benefits

Method	Dataset				Average
	Scene	Web	Document	Handwriting	
CRNN [26]	53.41 / 0.712	57.00 / 0.716	96.62 / 0.992	50.83 / 0.814	63.66 / 0.792
ASTER [27]	61.34 / 0.815	51.67 / 0.715	96.19 / 0.991	37.00 / 0.683	65.69 / 0.836
MORAN [21]	54.61 / 0.684	31.47 / 0.446	86.10 / 0.962	16.24 / 0.305	55.26 / 0.682
SAR [15]	59.67 / 0.766	58.03 / 0.716	95.67 / 0.988	36.49 / 0.736	65.07 / 0.811
SEED [23]	44.72 / 0.681	28.06 / 0.460	91.38 / 0.980	20.97 / 0.475	51.43 / 0.626
MASTER [20]	62.82 / 0.726	52.05 / 0.620	84.39 / 0.944	26.92 / 0.443	62.39 / 0.773
ABINet [8]	66.55 / 0.792	63.17 / 0.776	98.19 / 0.996	53.09 / 0.813	72.06 / 0.847
TransOCR [4]	71.33 / 0.823	64.81 / 0.764	97.07 / 0.993	53.00 / 0.797	74.55 / 0.843
TransOCR + PRAB [4]	**71.60 / 0.834**	65.52 / 0.782	97.36 / 0.994	53.67 / 0.802	74.91 / 0.852
Ours	71.31 / 0.829	**69.21 / 0.797**	**98.29 / 0.997**	**60.30 / 0.849**	**76.13 / 0.892**

Table 3. Comparison with previous methods on the CTR benchmark. LACC / NED follows the percentage and decimal format, respectively.

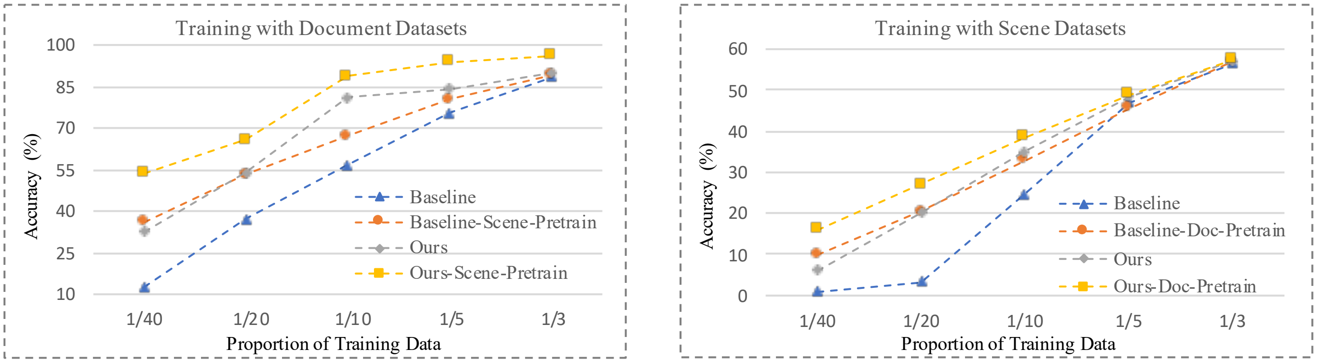


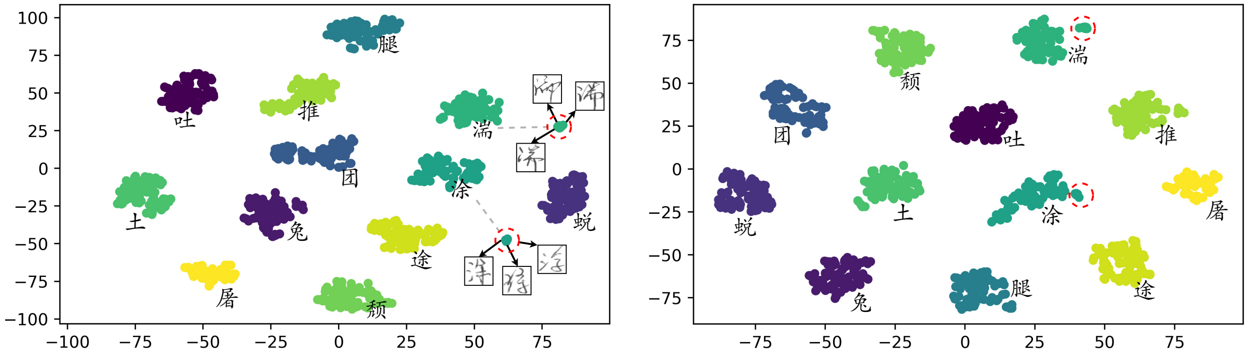
Figure 7. Performance comparison in data-scarce situations. “Scene-Pretrain” and “Doc-Pretrain” indicate that the model is pre-trained on the scene and document datasets, respectively. The proposed method performs better when the same strategy is adopted.

from additional template character images during training. Moreover, the CCR-CLIP model outperforms radical-based methods [2, 32, 31] with less inference time. However, the experimental results obtained on the scene character dataset CTW suggest that there is still much room for existing methods to further improve in performance, as the samples in CTW often suffer from severe occlusion and blurring problems, which indeed poses difficulties to CCR methods. We also conduct experiments to evaluate the time efficiency of the proposed method for a comprehensive comparison with existing methods. To ensure fairness, we set the batch size to 32 and calculate the average inference time for 200 batches during the test stage. As shown in Table 2, the CCR-CLIP method exhibits higher time efficiency than decomposition-based CCR methods.

4.2. Results on Chinese Text Recognition

We conduct experiments on a recently proposed benchmark for Chinese text recognition [6], which contains four types of datasets: scene, web, document, and handwriting. The experimental results reported in Table 3 demonstrate that the proposed two-stage CTR method outperforms previous methods by a clear margin on the web, document, and handwriting datasets. We also evaluate the recognition accuracy of zero-shot and few-shot (1-50 shots) characters on the test sets of four types (see Figure 6). Benefiting from the design of the proposed image-IDS matching framework, our method can easily recognize zero-shot and few-shot Chinese characters. The results indicate that the proposed method performs much better than the baseline model TransOCR [4] in both zero-shot and few-shot character settings. Specifically, our method achieves an accuracy of 51.4% in average in the zero-shot character setting, while TransOCR cannot recognize them at all. In the few-shot character setting, the proposed method achieves an improvement of 10.6% in average across the four types of datasets. However, we observe that the performance of our method is subpar on the scene dataset. A possible reason is that around 1/5 of the samples in the training set are vertical, which poses difficulties for our method.

In practical applications, collecting a large amount of annotated training data for the target domain is difficult and time-consuming. To further explore the effectiveness of our method in the case of limited training data, we randomly select subsets from the training data of the scene and document types. As shown in Figure 7, when using the same training strategy, our method outperforms the baseline model TransOCR by a clear margin on both scene and document datasets. This validates the robustness of our method in data-scarce situations.



(a) Without the loss $\mathcal{L}_I$

(b) With the loss $\mathcal{L}_I$

Figure 8. Character distribution visualization of whether introducing the loss $\mathcal{L}_I$ into the proposed CCR-CLIP model. The samples in red circles in (a) represent outliers with incorrect predictions, and the corresponding samples are also marked by red circles in (b). The gray lines connect the outliers and the class centers that they should correspond to.

MH	RT	Scene	Web	Document	Handwriting
		71.33%	64.81%	97.07%	53.00%
✓		70.17%	67.95%	97.97%	58.54%
✓	✓	71.31%	**69.21%**	**98.29%**	**60.30%**

Table 4. Results of ablation study. “MH” and “RT” denote the matching head and the regularization term in $\mathcal{L}_{ctr}$

Dataset	Character	Radical	Stroke
HWDB	96.83%	**97.18%**	92.74%
CTW	82.73%	**85.78%**	83.25%

Table 5. Comparison between different level representations.

4.3. Ablation Study

To evaluate the performance gain of the proposed matching head and the regularization term in $\mathcal{L}_{ctr}$, we conduct ablation experiments on them. According to the experimental results in Table 4, the proposed matching head results in 3.14%, 0.90%, and 5.54% performance gains on the web, document, and handwriting datasets, respectively. When introducing the regularization term, the proposed method further achieves an improvement of around 1.12% in average on the four datasets.

5. Discussions

Decomposition Levels. As introduced in Section 2.1, a Chinese character has three types of representations. In the proposed CCR-CLIP model, each type of representation can be fed into the text decoder to extract specific information for each Chinese character. To select the most effective representation for the text encoder, we conduct corresponding experiments. The results reported in Table 5 indicate that the CCR-CLIP model achieves the best performance when the radical-level representation is adopted. The relative poor performance of the stroke-level representation could be attributed to the fact that strokes are too fine-grained to perceive. Therefore, we choose the radical-level representation as the input of the text encoder.

Visualization. In order to validate the effectiveness of $\mathcal{L}_I$, we sample 1,200 handwritten examples of 12 characters from ICDAR2013 [36] and visualize the embedded visual features in a 2-D space with t-SNE, where each character class is denoted by one color. As shown in Figure 8(a), some scribbled character samples are far away from the corresponding cluster center in the feature space, which results in incorrect predictions. When $\mathcal{L}_I$ is introduced, most of the scribbled character samples are correctly predicted and closer to their cluster centers (see Figure 8(b)), which validates the effectiveness of $\mathcal{L}_I$ in the proposed CCR-CLIP. More visualization results and failure cases are shown in the Supplementary Material.

Limitations. In the proposed method, we incorporate a pre-processing step that the text images are rotated by 90 degrees anticlockwise if they are in a vertical orientation. Since the proposed method is based on canonical representation matching, the features of the same character in different orientations may cause confusion to the model. This may explain why the performance of our method is subpar on the scene dataset.

6. Conclusion

In this paper, we propose a novel two-stage framework for Chinese text recognition, which is inspired by the way humans recognize Chinese texts. The first stage involves a CCR-CLIP model that learns canonical representations of Chinese characters by aligning printed character images and Ideographic Description Sequences (IDS). In the second stage, using the learned canonical representations as supervision, we train a Chinese text recognition model with an image-IDS matching head. Extensive experiments demonstrate that the proposed method outperforms previous SOTA methods in both Chinese character recognition and Chinese text recognition tasks.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No.62176060), STCSM projects (No.20511100400, No.22511105000), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103), and the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

References

[1] Xiang Ao, Xu-Yao Zhang, and Cheng-Lin Liu. Cross-modal prototype learning for zero-shot handwritten character recognition. *Pattern Recognition*, 131:108859, 2022.

[2] Zhong Cao, Jiang Lu, Sen Cui, and Changshui Zhang. Zero-shot handwritten chinese character recognition with hierarchical decomposition embedding. *Pattern Recognition*, 107:107488, 2020.

[3] Fu Chang. Techniques for solving the large-scale classification problem in chinese handwriting recognition. In *Summit on Arabic and Chinese Handwriting Recognition*, pages 161–169. Springer, 2006.

[4] Jingye Chen, Bin Li, and Xiangyang Xue. Scene text telescope: Text-focused scene image super-resolution. In *CVPR*, 2021.

[5] Jingye Chen, Bin Li, and Xiangyang Xue. Zero-shot chinese character recognition with stroke-level decomposition. *IJCAI*, 2021.

[6] Jingye Chen, Haiyang Yu, Jianqi Ma, Mengnan Guan, Xixi Xu, Xiaocong Wang, Shaobo Qu, Bin Li, and Xiangyang Xue. Benchmarking chinese text recognition: Datasets, baselines, and an empirical study. *arXiv preprint arXiv:2112.15093*, 2021.

[7] Dan Cireşan and Ueli Meier. Multi-column deep neural networks for offline handwritten chinese character classification. In *2015 international joint conference on neural networks (IJCNN)*, pages 1–6. IEEE, 2015.

[8] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *CVPR*, 2021.

[9] Likun Gao, Heng Zhang, and Cheng-Lin Liu. Regularizing ctc in expectation-maximization framework with application to handwritten text recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2021.

[10] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[13] Lisheng Jin, Huacai Xian, Jing Bie, Yuqin Sun, Haijing Hou, and Qingning Niu. License plate recognition algorithm for passenger cars in chinese residential areas. *Sensors*, 12(6):8355–8370, 2012.

[14] Lian-Wen Jin, Jun-Xun Yin, Xue Gao, and Jiang-Cheng Huang. Study of several directional feature extraction methods with local elastic meshing technology for hccr. In *Proceedings of the Sixth Int. Conference for Young Computer Scientist*, pages 232–236, 2001.

[15] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, 2019.

[16] Zhiyuan Li, Qi Wu, Yi Xiao, Min Jin, and Huaxiang Lu. Deep matching network for handwritten chinese character recognition. *Pattern Recognition*, 107:107471, 2020.

[17] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[18] Chang Liu, Chun Yang, and Xu-Cheng Yin. Open-set text recognition via character-context decoupling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4523–4532, 2022.

[19] Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang. Online and offline handwritten chinese character recognition: benchmarking on new databases. *Pattern Recognition*, 46(1):155–162, 2013.

[20] Ning Lu, Wenwen Yu, Xianbiao Qi, Yihao Chen, Ping Gong, Rong Xiao, and Xiang Bai. Master: Multi-aspect non-local network for scene text recognition. *Pattern Recognition*, 117:107980, 2021.

[21] Canjie Luo, Lianwen Jin, and Zenghui Sun. Moran: A multi-object rectified attention network for scene text recognition. *Pattern Recognition*, 2019.

[22] Guo-Feng Luo, Da-Han Wang, Xia Du, Hua-Yi Yin, Xu-Yao Zhang, and Shunzhi Zhu. Self-information of radicals: A new clue for zero-shot chinese character recognition. *Pattern Recognition*, 140:109598, 2023.

[23] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *CVPR*, 2020.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[25] Xiaohang Ren, Yi Zhou, Zheng Huang, Jun Sun, Xiaokang Yang, and Kai Chen. A novel text structure feature extractor for chinese scene text detection and recognition. *IEEE Access*, 5:3193–3204, 2017.

[26] Baoguang Shi, Xiang Bai, and Cong 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*, 2016.

[27] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[28] Yih-Ming Su and Jhing-Fa Wang. A novel stroke extraction method for chinese characters using gabor filters. *Pattern Recognition*, 36(3):635–647, 2003.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[30] Zhaoyi Wan, Fengming Xie, Yibo Liu, Xiang Bai, and Cong Yao. 2d-ctc for scene text recognition. *arXiv preprint arXiv:1907.09705*, 2019.

[31] Tianwei Wang, Zecheng Xie, Zhe Li, Lianwen Jin, and Xiangle Chen. Radical aggregation network for few-shot offline handwritten chinese character recognition. *Pattern Recognition Letters*, 125:821–827, 2019.

[32] Wenchao Wang, Jianshu Zhang, Jun Du, Zi-Rui Wang, and Yixing Zhu. Denseran for offline handwritten chinese character recognition. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 104–109. IEEE, 2018.

[33] Yao Xiao, Dan Meng, Cewu Lu, and Chi-Keung Tang. Template-instance loss for offline handwritten chinese character recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 315–322. IEEE, 2019.

[34] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3474–3482, 2018.

[35] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C Lee Giles. Improving offline handwritten chinese character recognition by iterative refinement. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 5–10. IEEE, 2017.

[36] Fei Yin, Qiu-Feng Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Icdar 2013 chinese handwriting recognition competition. In *2013 12th international conference on document analysis and recognition*, pages 1464–1470. IEEE, 2013.

[37] Haiyang Yu, Jingye Chen, Bin Li, and Xiangyang Xue. Chinese character recognition with radical-structured stroke trees. *arXiv preprint arXiv:2211.13518*, 2022.

[38] Tai-Ling Yuan, Zhe Zhu, Kun Xu, Cheng-Jun Li, Tai-Jiang Mu, and Shi-Min Hu. A large chinese text dataset in the wild. *Journal of Computer Science and Technology*, 34(3):509–521, 2019.

[39] Jianshu Zhang, Jun Du, and Lirong Dai. Radical analysis network for learning hierarchies of chinese characters. *Pattern Recognition*, 103:107305, 2020.

[40] Yuanzhi Zhu, Zecheng Xie, Lianwen Jin, Xiaoxue Chen, Yaoxiong Huang, and Ming Zhang. Scut-ept: New dataset and benchmark for offline chinese text recognition in examination paper. *IEEE Access*, 7:370–382, 2018.

[41] Xinyan Zu, Haiyang Yu, Bin Li, and Xiangyang Xue. Chinese character recognition with augmented character profile matching. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6094–6102, 2022.