Benchmarking Chinese Text Recognition: Datasets, Baselines, and an Empirical Study

Jingye Chen, Haiyang Yu, Jianqi Ma, Mengnan Guan, Xixi Xu, Xiaocong Wang, Shaobo Qu, Bin Li*, Xiangyang Xue*
Shanghai Key Laboratory of Intelligent Information Processing
School of Computer Science, Fudan University
{jingyechen19, hyyu20, majq16, libin, xyxue}@fudan.edu.cn,

Abstract

The flourishing blossom of deep learning has witnessed the rapid development of text recognition in recent years. However, the existing text recognition methods are mainly for English texts, whereas ignoring the pivotal role of Chinese texts. As another widely-spoken language, Chinese text recognition in all ways has extensive application markets. Based on our observations, we attribute the scarce attention on Chinese text recognition to the lack of reasonable dataset construction standards, unified evaluation methods, and results of the existing baselines. To fill this gap, we manually collect Chinese text datasets from publicly available competitions, projects, and papers, then divide them into four categories including scene, web, document, and handwriting datasets. Furthermore, we evaluate a series of representative text recognition methods on these datasets with unified evaluation methods to provide experimental results. By analyzing the experimental results, we surprisingly observe that state-of-the-art baselines for recognizing English texts cannot perform well on Chinese scenarios. We consider that there still remain numerous challenges under exploration due to the characteristics of Chinese texts, which are quite different from English texts. The code and datasets are made publicly available at https://github.com/FudanVI/benchmarking-chinese-text-recognition.

Figure 1. Three reasons for the scarce attention of Chinese text recognition. (a) People may use different ways to crop text regions, which leads to unfair comparison. (b) It is necessary to specify the equivalence between lowercase and uppercase, half-width and full-width, simplified and traditional characters. (c) The existing methods are mainly evaluated with English datasets rather than Chinese datasets.

1. Introduction

In recent years, text recognition has attracted extensive attention due to its wide applications such as autonomous driving [53, 92], document retrieval [49, 85], signature identification [52, 55], etc. However, the existing text recognition methods mainly focus on English texts [4, 15, 16, 27, 36, 45, 54, 58–60, 65, 66] while ignoring the huge market of Chinese text recognition (CTR). Specifically, Chinese is the most spoken language across the world with 1.31 billion speakers, meaning that CTR and its downstream tasks will certainly have crucial impacts on this population. Based on our observations, we summarize three crucial reasons for the scarce attention of Chinese text recognition:
Lack of reasonable dataset construction standards. Although there are some publicly available CTR datasets provided by the relevant competitions, some inconsistent issues regarding dataset construction standards still remain. For example, some competitions (e.g., RCTW [61] and ReCTS [96]) tend to evaluate the performance of text spotters and only provide global images, points of bounding boxes, and text labels. On the contrary, current mainstream recognition methods mainly rely on the cropped text regions as input. Ideally, based on the given quadrangle boxes, we can prepare the input for recognizers by cropping the text regions along the points and then rectify it to a horizontal-oriented image (see the left branch of Figure 1 (a)), which can effectively eliminating useless background areas compared to those methods directly using the minimum circumscribed horizontal boxes (see the right branch of Figure 1 (a)). However, as we observe from the datasets, the starting points of a large amount of annotating areas are not always indicating the same corner of the bounding boxes (e.g., top-left) due to incorrect labeling, leading to rectification failure like laying the characters down to the left or right, or turning it upside down, which indeed confuse the CTR methods. Consequently, these difficulties will force people to pick the cropping strategy that benefits the final performance, leading to unfair comparison incurred by inconsistent input. Also, different datasets are collected under different environments, in which the data appearance shows various distributions with the environments. Finding a reasonable splitting strategy is also beneficial for us to perform more effective research. Hence, the standard of dataset construction should be seriously taken into consideration.

Lack of unified evaluation methods. Generally, to evaluate the text recognition methods on English texts, we usually transfer the uppercase to lowercase by default. However, there do not exist any unified evaluation methods to fairly assess the CTR methods. Practically, we observe that two forms of characters, including full-width and half-width characters (see Figure 1(b)), may simultaneously appear in the labels. In this case, the dataset users may feel confused that whether they should be equally treated as one character in the alphabet. Besides, it is worth discussing that whether characters in traditional form and simplified form should be viewed equally. Therefore, unified evaluation methods are urgently needed to fairly assess the CTR methods. Additionally, the evaluation metrics are inconsistent across CTR papers (e.g., normalized edit distance used in [96] but accuracy rate used in [93]), which is inconvenient for further comparisons.

Lack of experimental results of the existing baselines. On one hand, the existing text recognition methods are mainly evaluated on English texts like IIIT5K [50], IC03 [44], IC13 [34], IC15 [33], CT80 [56], etc. Although few methods attempt to experiment on Chinese text datasets, the details regarding dataset construction are not clear in the corresponding papers, which makes it difficult for other people to use as CTR baselines (see Figure 1(c)). On the other hand, reproducing the results of the existing text recognition methods for constructing CTR baselines is a laborious task. It not only takes a large quantity of time but also consumes numerous GPU resources, which indeed lessens researchers’ passion for Chinese text recognition.

In this article, we seek to construct a benchmark for Chinese text recognition to fill this gap. We first derive the existing Chinese text datasets from the public competitions, papers, and projects, resulting in four categories, including scene, web,
document, and handwriting (see Figure 2). Then we manually divide each dataset into training, validation, and testing sets at a reasonable ratio. Please note that distinct from the existing English text benchmarks, we specially design the validation set to fairly compare the existing methods, i.e., making sure that the best model is chosen base on the validation set rather than the testing set. Besides, we reproduce the results of some representative text recognition methods (e.g., CRNN [59], ASTER [60], MORAN [45], etc.) on the collected dataset as baselines in our benchmark. By analyzing the experimental results of the baselines, we observe that some state-of-the-art methods originally designed for English texts cannot well adapt to the Chinese texts. Through in-depth analysis, we observe that some characteristics of Chinese texts (e.g., long text, large alphabet, complicated semantic features) indeed put obstacles to the existing methods in different ways. We also demonstrate the rationality of the dataset construction in terms of each category’s recognizability which is calibrated by humans. Overall, the contributions of this article can be listed as follows:

• We manually collect Chinese text recognition datasets from public competitions, papers, and projects. Then we divide them into four categories, including scene, web, document, and handwriting datasets. We further split these datasets into training, evaluation, and testing sets at a reasonable ratio.

• We propose some evaluation methods to fairly compare the existing text recognition methods.

• Based on the collected datasets and proposed evaluation methods, we reproduce the results of a series of baselines, then analyze the performance of the baselines in detail.

2. Preliminaries

In this section, we first introduce the hierarchical representations for Chinese characters, then discuss some characteristics of Chinese texts that are distinct from English texts. Finally, we introduce the existing text recognition methods.

2.1. Hierarchical Representations for Chinese Characters

Here we introduce three ways to represent Chinese characters (see the examples “奇” and “绍” in Figure 3(a)), including character level, radical level, and stroke level.

Character level. According to Chinese national standard GB18030-2005, the number of Chinese characters is 70,244 in total, where 3,755 characters are Level-1 commonly-used characters.

Radical level. According to the unicode standards for ideographic description characters, there are 12 radical structures (see Figure 3(b)) and 514 radicals in the Level-1 commonly-used Chinese characters. For the 3,755 commonly-used Chinese characters, the radical-level representation can effectively reduce the size of alphabet from 3,755 to 526.

Stroke level. According to Unicode Han Database, each Chinese character can be decomposed into a stroke sequence. There are five basic categories of strokes (e.g., horizontal, vertical, left-falling, right-falling, and turning), each of which contains several instances (see Figure 3(c)).

---

1 https://zh.wikipedia.org/wiki/GB_18030
2 https://unicode.org/charts/PDF/U2FF0.pdf
3 http://www.unicode.org
2.2. Characteristics of Chinese Texts

It is commonsense in the community that Chinese texts are harder to recognize compared with English texts [11, 74]. To explore the intrinsic reasons, here we analyze the characteristics of Chinese texts that are distinct from English texts:

**Large amount of characters.** According to the national standard GB18030-2005, the number of Chinese characters is 70,244 (including 3,755 commonly-used Level-1 characters). It is much larger than the scale of English characters, which only contains 26 uppercase and 26 lowercase letters (see Figure 4(a)). On one hand, performing large-scale classification is essentially a difficult task [18]. One the other hand, the recognizers are likely to encounter challenging zero-shot problems i.e., the characters to be tested may be absent from the training set.

**Similar appearance.** Compared with English letters, there are considerable Chinese character clusters with similar appearance. As shown in Figure 4(b), the difference of character pairs “戌” and “戍” only lie in a tiny strokes. They are hard to distinguish even for human eyes, which indeed burdens the existing text recognition methods.

**Complex sequential features.** We observe that most samples in the existing English benchmark are existing words. Since there is a space between two English words, humans tend to split them apart during the labeling process for detection. Besides, there are inherent statistical features within English words (e.g., “abl” is more likely followed by “e”), thus helping recognizers better capture the sequential features. On the contrary, Chinese texts are more likely to appear in phrases or sentences. Under this circumstance, there are complicated dependencies between Chinese characters in terms of the part of speech (see Figure 4(c)), which indeed bring difficulties for the recognizers to learn the sequential features.

**Commonly-seen vertical texts.** Compared with English texts, Chinese texts are more likely to appear vertically due to the commonly-used traditional couplets or signboards in natural scenes (see Figure 4(d)). On the contrary, there are few vertical English texts due to people’s inherent reading habits.

In general, these characteristics of Chinese texts suggest that the model design and learning for Chinese text recognition can be distinct from the way for English text recognition.

2.3. Existing Text Recognition Methods

**Generic text recognition methods.** To tackle the text recognition problems, the traditional methods [1, 57, 68, 69] are mainly branched into two parts, i.e., the top-down and bottom-up approaches. The top-down approach considers each word as a unique category and performs word-level classification, while the bottom-up approach perceives each character as a category by performing character-level recognition and then grouping them into words. Jaderberg et al. [31] put forward a representative top-down method to exploit word-level deep features of text images to perform word-level classification for the 90k common-used words. Early bottom-up methods like PhotoOCR [7] and Tesseract [62] tended to locate the exact text region using image processing operations and perform character recognition using SVM or other linear classifiers on segmented regions, and finally group the separated character pieces into a whole word.

As the thriving of deep learning, the researchers also made attempts to build the text recognition models based on deep neural networks following the bottom-up fashion [2,3,5,6,22,23,29,35,38,40–43,46,47,51,59,60,65,67,71,73,83,87,90,98, 100]. For example, CRNN [59] utilizes the CNN-RNN architecture to extract features for the text images, which are further supervised with the CTC loss [24] to maximize the probability of the ground truth. However, as CRNN restricts the height of features to 1 for decoding, it is vulnerable to some distorted text images in natural scenes. In this case, some rectification-based methods [30,45,60,91] are put forward to mitigate this problem. For instance, ASTER [60] takes advantage of Spatial Transformer Networks [32] to rectify the distorted texts with control points. MORAN [45] first generates the pixel-wise offset maps in a weakly-supervise manner, then rectifies the text images through pixel sampling. In addition to the rectification-
based methods, other methods attempt to tackle text recognition in the 2-D feature domain \cite{36, 37}. For example, SAR \cite{36} directly utilizes the 2-D features for decoding, and Liao et al. \cite{37} took advantage of the character-level annotation to predict the segmentation mask for each character with 2-D feature maps. Also, some works \cite{21, 58, 82} attempt to construct text recognizers based on Transformer \cite{64}, which is thrived in the field of natural language processing, to robustly learn the representations for text images through self-attention modules. Recently, some researchers incorporated semantic knowledge into text recognizers to fully exploit the external language priors \cite{20, 54, 75, 88}. For example, SEED \cite{54} utilizes text embedding guided by fastText \cite{8} to initialize the attention-based decoder. ABINet \cite{20} designs two autonomous branches to iteratively optimize the vision and language models. Overall, these general text recognition methods can be easily transferred to Chinese scenarios by simply replacing the alphabet or language priors. Nevertheless, their performance still has a large room for improvement if the characteristics of Chinese texts are further taken into account.

**Chinese text recognition methods.** According to our observations, only few methods are specifically designed for Chinese text recognition although it has enormous application markets all over the world. In particular, the existing Chinese text methods mainly aim at handwriting text recognition \cite{13, 14, 19, 28, 76, 79} or handwriting character recognition \cite{9, 11, 77, 80, 84, 94, 95, 97} thanks to the detailed specifications formulated by the CASIA-HWDB database \cite{39}. For example, Xiao et al. \cite{81} put forward a template-instance loss to distinguish the similar Chinese characters pairs in the feature domain. Xiao et al. \cite{86} utilized an iterative attention mechanism to progressively concentrate on the distinguishable regions of Chinese characters. In \cite{78}, the authors leveraged the Hidden Markov Model to tackle Chinese text recognition, while exploiting the similarity of Chinese characters to reduce the number of total hidden states. However, these recognition methods all work at the character level and they are vulnerable to those characters absent from the training sets. To mitigate the character zero-shot problems, some researchers sought to tackle unseen Chinese characters at the radical level \cite{9, 70, 74, 94}. For example, DenseRAN \cite{74} follows the fashion of image captioning and decodes the radical sequence iteratively. In \cite{70}, the authors utilize the printed images as support samples to enforce the same radical to be close in the feature domain. Although these radical-based methods can alleviate the character zero-shot problem to some extent, another dilemma named the radical zero-shot problem may arise if some radicals have not appeared in the training sets. To fundamentally resolve the zero-shot problems, Chen et al. \cite{11} tried to decompose characters into strokes, which are the atomic units of Chinese characters. Similarly, the authors utilized the iterative manner to generate the stroke sequences of a given character, then leveraged a Siamese architecture to resolve the one-to-many problem guided by visual features.

Overall, the research in Chinese text recognition is still far from satisfying due to the following two reasons: (1) Inconsistent dataset construction methods and experimental settings across the existing papers; (2) Current Chinese text recognition methods are mainly tested using the handwriting datasets, whereas lacking the experiments on other datasets (e.g., scene, web, and document datasets), which are also commonly-seen in our daily lives. Under this circumstance, the datasets and baselines are required to be constructed and evaluated with reasonable and unified standards.

3. Datasets

In this section, we first introduce the ways to preprocess the Chinese text recognition datasets, then detail datasets of each categories (e.g., scene, web, document, and handwriting). At last, we have some analysis on the collected datasets.

3.1. Preprocessing

Here we carefully design four steps to preprocess the datasets:

**1. Reserve the text images that contain other languages.** We observe that the Chinese text recognition datasets mainly comprises Chinese characters, meanwhile containing a few English characters as well as other languages (e.g., Japanese and Korean). Considering the language distributions of text images in natural scenes, we decide to reserve the samples with characters in other languages.

**2. Remove the samples annotated as “###”.** According to the labeling standards of some datasets (e.g., RCTW \cite{61} and ReCTS \cite{96}), the illegible text images are annotated as “###” (see Figure 5). We observe that there exist heavy blur or occlusion within these samples, which are hard to recognize even for human eyes. Considering these samples may bring noises to the training process, we decide to remove them from the datasets.

**3. Highlight the importance of validation sets.** Through observations, the existing English text recognition benchmarks usually lack the validation sets. Practically, the testing sets are not available during the training stage and the best model should be chosen according to the performance on the validation sets. Here we take the validation sets into account in pursuit of more rigorous research.
(4) Only utilize those samples with available text labels. We observe that the text labels of many testing sets are not publicly available, especially for those competition datasets. Although we can submit the text predictions on the corresponding websites, it is inconvenient as some websites limit the maximum submission times per day. Under this circumstance, we only use those samples with available annotated text labels.

Based on these principles, we collect four categories of datasets from available resources, which will be introduced next.

3.2. Details of Datasets

Scene dataset. From the competitions, papers, and projects, we have derived a series of scene subdatasets, including RCTW [61], ReCTS [96], LSVT [63], ArT [17], and CTW [89]. The details of each subdataset are introduced as follows:

RCTW [61]: It provides 12,263 annotated Chinese text images from natural scenes. We derive 44,420 text lines from the training set and use them in our benchmark. The testing set of RCTW is not used as the text labels are not available.

ReCTS [96]: It provides 25,000 annotated street-view Chinese text images, mainly derived from natural signboards. We only adopt the training set and crop 107,657 text samples in total for our benchmark.

LSVT [63]: It is a large scale Chinese and English scene text dataset, providing 50,000 full-labeled (polygon boxes and text labels) and 400,000 partial-labeled (only one text instance each image) samples. We only utilize the full-labeled training set and crop 243,063 text line images for our benchmark.

ArT [17]: It contains text samples captured in natural scenes with various text layouts (e.g., rotated text and curved texts). Here we obtain 49,951 cropped text images from the training set, and use them in our benchmark.

CTW [89]: It contains annotated 30,000 street view images with rich diversity including planar, raised, and poorly-illuminated text images. Also, it provides not only character boxes and labels, but also character attributes like background complexity, appearance, etc. Here we crop 191,364 text lines from both the training and testing sets.

We combine all the subdatasets, resulting in 636,455 text samples. We randomly shuffle these samples and split them at a ratio of 8:1:1, leading to 509,164 samples for training, 63,645 samples for validation, and 63,646 samples for testing.

Web dataset. To collect the web dataset, we utilize MTWI [26] that contains 20,000 Chinese and English web text images from 17 different categories on the Taobao website. The text samples are appeared in various scenes, typography and designs. We derive 140,589 text images from the training set, and manually divide them at a ratio of 8:1:1, resulting in 112,471 samples for training, 14,059 samples for validation, and 14,059 samples for testing.

Document dataset. We use the public repository Text Render\(^4\) to generate some document-style synthetic text images. More specifically, we uniformly sample the length of text varying from 1 to 15. The corpus comes from wiki, films, amazon, and baike. The dataset contains 500,000 in total and is randomly divided into training, validation, and testing sets with a proportion of 8:1:1 (400,000 v.s. 50,000 v.s. 50,000).

Handwriting dataset. We collect the handwriting dataset based on SCUT-HCCDoc [93], which captures the Chinese handwritten image with cameras in unconstrained environments. Following the official settings, we derive 93,254 text lines for training and 23,389 for testing, respectively. To pursue more rigorous research, we manually split the original training set into two sets at a ratio of 4:1, resulting in 74,603 samples for training and 18,651 samples for validation. For convenience, we continue to use the original 23,389 samples for testing.

\(^4\)https://github.com/Sanster/text_renderer
3.3. More Analysis of Datasets

Analysis of the alphabet size and the amount of characters. Referring to Table 1, we can see that the alphabet size and the proportion of Chinese characters vary across the four datasets. For example, most of the web texts are Chinese advertisements with fixed expressions for customers around the world, thus containing fewer characters in the alphabet (4,402 characters). Besides, the web dataset comprises many telephone numbers and English websites, leading to the least proportion of Chinese characters among the four datasets (44.9%). For the handwriting dataset, most texts are copied from Chinese ancient poems, which contain more rarely-used characters than other datasets, thus resulting in the largest alphabet (6,105 characters). Besides, the testing set of the handwriting dataset contains the most zero-shot characters (253 zero-shot characters) that are absent from the training set, which further increases the difficulty for recognition.

Analysis of the distribution of text length and aspect ratio Figure 6 and Figure 7 illustrate the distribution of text length and aspect ratio (i.e., width to height). From these figures, we observe that long texts (e.g., length ≥ 10) appear more frequently in the handwriting dataset, which brings difficulties to the baselines. On the contrary, the text in the scene and web datasets are relatively short, perhaps considering the reading efficiency of passengers, customers, etc. For the ratio distribution, we observe that the scene dataset has more vertical text than others (ratio ≤ 1) due to the commonly-used couplets and signboards in the wild. By contrast, the handwriting dataset comprises more horizontal texts.

Analysis of the character and word frequency. As shown in Table 2, we analyze the frequency of characters and words in each dataset and observe some interesting phenomenons. For instance, as there are many Chinese brands or state-owned enterprises in natural scenes, the characters “中”, “国”, and the word “中国” appear frequently in the scene dataset. Additionally, the web dataset contains many high-frequency advertising terms like “包邮”, “正品”, etc. For the document and handwriting datasets, the particle “的” appears most frequently. In particular, in the handwriting dataset, since some text images are cropped from diaries, there are many characters or words referring to people like “我”, “你”, “我们”, etc.

Recognizability calibration by humans. To figure out the factors that impair the recognizability of text images in the four datasets, we invite 20 highly-educated participants to conduct this experiment. We invite the participants to identify the corresponding reasons (multiple choices) that hamper the recognizability: (1) Occlusion (occluded by other objects and

---

Table 1. The statistical results of the alphabet size and the amount of characters. “Chinese”, “All”, “Proportion” denote the number of Chinese characters, all characters, the proportion of Chinese characters, respectively.

| Dataset    | Training Samples | Validation Samples | Testing Samples | Alphabet Size | Zero-Shot Characters | The Amount of Characters | Chinese | All    | Proportion |
|------------|------------------|--------------------|-----------------|---------------|---------------------|-------------------------|---------|--------|------------|
| Scene      | 509,164          | 63,645             | 63,646          | 5,880         | 109                 | 2,188,014               | 3,207,396| 68.2%  |
| Web        | 112,471          | 14,059             | 14,059          | 4,402         | 94                  | 315,587                 | 703,256 | 44.9%  |
| Document   | 400,000          | 50,000             | 50,000          | 4,865         | 25                  | 3,386,138               | 3,997,505| 84.7%  |
| Handwriting| 74,603           | 18,651             | 23,389          | 6,105         | 253                 | 983,989                 | 1,154,160| 85.3%  |

---

5 We use jieba to split texts into words. The homepage of jieba: https://github.com/fxsjy/jieba
6 Ensure that the participant can recognize all well-written characters.
Text recognition has achieved rapid progress over the last decade. According to the major characteristics, text recognition methods can be classified into several categories, including CTC-based methods, rectification-based methods, Transformer-based methods, etc. From these categories, we manually select seven representative methods as baselines (the detailed characteristics are shown in Table 4), which will be introduced as follows.

CRNN (Shi et al., 2016) [59] is a typical CTC-based method and it is widely used in academia and industry. It first sends the text image to a CNN to extract the image features, then adopts a two-layer LSTM to encode the sequential features. Finally, the output of LSTM is fed to a CTC (Connectionist Temporal Classification) [24] decoder to maximize the probability of all the paths towards the ground truth.

ASTER (Shi et al., 2018) [60] is a typical rectification-based method aiming at tackling irregular text images. It introduces a Spatial Transformer Network (STN) [32] to rectify the given text image into a more recognizable appearance. Then the rectified text image is sent to a CNN and a two-layer LSTM to extract the features. In particular, ASTER takes advantage of the attention mechanism to predict the final text sequence.

MORAN (Luo et al., 2019) [45] is a representative rectification-based method. It first adopts a multi-object rectification network (MORN) to predict rectified pixel offsets in a weak supervision way (distinct from ASTER that utilizes STN). The output pixel offsets are further used for generating the rectified image, which is further sent to the attention-based decoder (ASRN) for text recognition.
Table 4. The characteristics of the baselines used in our benchmark. The red checks indicate the most prominent characteristics.

| Baseline   | Rectification | Encoder           | Decoder                   |
|------------|---------------|-------------------|---------------------------|
|            |               | 1-D feature | 2-D feature | CTC | RNN | Transformer | Semantics-Enhanced |
| CRNN [59]  | ✓             | ✓              | ✓                        | ✓   | ✓   | ✓           | ✓                   |
| ASTER [60] | ✓             | ✓              | ✓                        | ✓   | ✓   | ✓           | ✓                   |
| MORAN [45] | ✓             | ✓              | ✓                        | ✓   | ✓   | ✓           | ✓                   |
| SAR [36]   | ✓             | ✓              | ✓                        | ✓   | ✓   | ✓           | ✓                   |
| SEED [54]  | ✓             | ✓              | ✓                        | ✓   | ✓   | ✓           | ✓                   |
| SRN [88]   | ✓             | ✓              | ✓                        | ✓   | ✓   | ✓           | ✓                   |
| TransOCR [10] | ✓     | ✓              | ✓                        | ✓   | ✓   | ✓           | ✓                   |

SAR (Li et al., 2019) [36] is a representative method that takes advantage of 2-D feature maps for more robust decoding. In particular, it is mainly proposed to tackle irregular texts. On one hand, SAR adopts more powerful residual blocks [25] in the CNN encoder for learning stronger image representation. On the other hand, different from CRNN, ASTER, and MORAN compressing the given image into a 1-D feature map, SAR adopts 2-D attention on the spatial dimension of the feature maps for decoding, resulting in a stronger performance in curved and oblique texts.

SEED (Qiao et al., 2020) [54] is a representative semantics-based method. It introduces a semantics module to extract global semantics embedding and utilize it to initialize the first hidden state of the decoder. Specifically, while inheriting the structure of ASTER, the decoder of SEED intakes the semantic embedding to provide prior for the recognition process, thus showing superiority in recognizing low-quality text images.

SRN (Yu et al., 2020) [88] is a representative semantics-based method that utilizes self-attention modules to correct the errors of predictions. It proposes a parallel visual attention module followed by a self-attention network to capture the global semantic features through multi-way parallel transmission, resulting in significant performance improvement towards the recognition of irregular texts.

TransOCR (Chen et al., 2021) [10] is one of the representative Transformer-based methods. It is originally designed to provide text priors for the super-resolution task. It employs ResNet-34 [25] as the encoder and self-attention modules as the decoder. Distinct from the RNN-based decoders, the self-attention modules are more efficient to capture semantic features of the given text images.

5. An Empirical Study

In this section, we first introduce the implementation details and evaluation methods, then we demonstrate the experimental results. Finally, we have detailed analysis of the experimental results and show some failure cases in each dataset.

5.1. Implementation Details

We adopt the off-the-shelf PyTorch implementations of CRNN [59], ASTER [60], MORAN [45], SAR [36], SEED [54], SRN [88], and TransOCR [10] on Github7 to reproduce the experimental results on the collected Chinese text images datasets. As demonstrated in Figure 7, the distributions of aspect ratios vary across different types of datasets. In this context, we empirically constrain the input size of scene, web, document, handwriting datasets to $64 \times 200$, $64 \times 200$, $64 \times 800$, and $64 \times 1200$, respectively, ensuring that the texts are recognizable for human eyes in most cases. We utilize the validation set of each dataset to determine the best models according to the recognition accuracy, then assess the baselines using the testing set. For convenience, we combine the alphabets of four datasets for all experiments, resulting in an overall alphabet of 7,934 characters. No other strategies like data augmentation, model ensembling, pre-training are used for the baselines.

5.2. Evaluation Methods

Practically, the unified evaluation methods are indispensable to fairly compare the baselines. Following the ICDAR2019 ReCTS Competition8, we propose some rules to convert the predictions and labels: (1) Convert full-width characters to

---

7CRNN: https://github.com/meijieru/crnn.pytorch, ASTER: https://github.com/ayumiymk/aster.pytorch, MORAN: https://github.com/Canjie-Luo/MORAN_v2, SEED: https://github.com/Pay20Y/SEED, SRN: https://github.com/PaddlePaddle/PaddleOCR, SAR: https://github.com/liuch37/sar-pytorch, TransOCR: https://github.com/FudanVI/benchmarking-chinese-text-recognition/tree/main/model

8https://rrc.cvc.uab.es/?ch=12&com=tasks
Table 5. The results of the baselines on four datasets. ACC / NED follow the percentage format and decimal format, respectively.

| Method    | Year | Scene       | Web         | Document     | Handwriting | Parameters |
|-----------|------|-------------|-------------|--------------|-------------|------------|
| CRNN [59] | 2017 | 53.4 / 0.734| 54.5 / 0.736| 97.5 / 0.994 | 46.4 / 0.840| 12.4M      |
| ASTER [60]| 2018 | 54.5 / 0.695| 52.3 / 0.689| 93.1 / 0.899 | 38.9 / 0.720| 27.2M      |
| MORAN [45]| 2019 | 51.8 / 0.686| 49.9 / 0.682| 95.8 / 0.991 | 39.7 / 0.761| 28.5M      |
| SAR [36]   | 2019 | 62.5 / 0.785| 54.3 / 0.725| 93.8 / 0.987 | 31.4 / 0.655| 27.8M      |
| SRN [88]   | 2020 | 60.1 / 0.778| 52.3 / 0.706| 96.7 /      | 18.0 / 0.512| 64.3M      |
| SEED [54]  | 2020 | 49.6 / 0.661| 46.3 / 0.637| 93.7 / 0.990 | 32.1 / 0.674| 73.5M      |
| TransOCR [10]| 2021| 53.4 / 0.802| 54.3 / 0.725| 93.8 / 0.987 | 31.4 / 0.655| 27.8M      |

Table 6. The results of the baselines on hard cases. ACC / NED follow the percentage format and decimal format, respectively.

| Methods     | Occlusion | Oblique or Curved | BG Confusion | Blur       | Long Text | Vertical Text |
|-------------|-----------|-------------------|--------------|------------|-----------|---------------|
| CRNN [59]   | 14.0 / 0.514 | 16.0 / 0.366     | 20.0 / 0.583 | 31.2 / 0.513 | 55.4 / 0.871 | 4.0 / 0.208   |
| ASTER [60]  | 31.0 / 0.505 | 18.0 / 0.313     | 26.0 / 0.460 | 39.6 / 0.503 | 44.6 / 0.740 | 10.0 / 0.164  |
| MORAN [45]  | 20.0 / 0.440 | 21.0 / 0.375     | 16.0 / 0.433 | 34.2 / 0.471 | 44.3 / 0.757 | 8.0 / 0.149   |
| SAR [36]    | 27.0 / 0.548 | 26.0 / 0.488     | 35.0 / 0.645 | 39.6 / 0.541 | 44.7 / 0.724 | 31.0 / 0.483  |
| SRN [88]    | 24.0 / 0.509 | 14.0 / 0.321     | 21.0 / 0.520 | 41.4 / 0.573 | 43.4 / 0.717 | 17.0 / 0.348  |
| SEED [54]   | 20.0 / 0.402 | 12.0 / 0.243     | 25.0 / 0.448 | 36.4 / 0.460 | 40.9 / 0.701 | 14.0 / 0.177  |
| TransOCR [10]| 34.0 / 0.625 | 31.0 / 0.560     | 38.0 / 0.694 | 43.4 / 0.570 | 61.4 / 0.885 | 40.0 / 0.566  |

half-width characters; (2) Convert traditional Chinese characters to simplified characters; (3) Convert uppercase letters to lowercase letters; (4) Remove all spaces.

After these transformations, we utilize two mainstream metrics, including Accuracy (ACC) and Normalized Edit Distance (NED), to evaluate the baselines. Specifically, the ACC and NED are calculated as follows:

\[
\text{ACC} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y_i = \hat{y}_i) 
\]

\[
\text{NED} = 1 - \frac{1}{N} \sum_{i=1}^{N} \frac{\text{distance}(y_i, \hat{y}_i)}{\maxlen(y_i, \hat{y}_i)} 
\]

where \(y_i\) and \(\hat{y}_i\) denote the \(i\)-th transformed prediction and the transformed label, respectively. \(\mathbb{I}\), “distance”, and “maxlen” denote the indication function, edit distance, and maximum sequence length, respectively. \(N\) is the number of text images. The ACC and NED are both ranged in \([0,1]\). Higher ACC and NED indicate better performance of the evaluated baseline.

5.3. Analysis of Experimental Results

Overall analysis. We have a comprehensive analysis on the performance of baselines on four datasets. The experimental results are demonstrated in Table 5.

Scene dataset. As shown in Table 3, the scene dataset is rather challenging due to some problems like occlusion, background confusion, and blur. Besides, as shown in Figure 7, the scene dataset comprises many vertical text images, which indeed put obstacles to those baselines (e.g., CRNN [59], ASTER [60], and MORAN [45]) that simply transform original input to 1-D feature sequence. By contrast, those 2-D methods (e.g., SAR [36], SRN [88] and TransOCR [10]) achieve better performance on this dataset, as 2-D feature maps are more robust to tackle text images with special layouts (e.g., vertical or curved). Further, by taking advantage of the self-attention modules, TransOCR [10] surpasses all its counterparts with recognition accuracy 63.3% as it is capable of modeling the sequential features more flexibly. At last, we notice that SEED [54] does not perform well on this dataset. We speculate that SEED needs to map each text image to its corresponding semantics embedding guided by fastText [8], whereas Chinese text usually contains complex semantic features, thus bringing difficulties to the semantic learning procedure.
Web dataset. Referring to Table 3, the web dataset is relatively easy to distinguish than the scene datasets. Even so, we notice that the performance of all baselines is lower than that in the scene dataset, which may stem from the scarcity of training samples (112,471 training samples for web v.s. 509,164 samples for scene). Moreover, we notice that the performance of CRNN [59], ASTER [60], SAR [36], SRN [88] do show much difference (i.e., fluctuate around 53). Meanwhile, the proportion of vertical text images is smaller (see Figure 7) than that of the scene dataset so the choices of 1-D or 2-D feature maps do not lead to essential differences. At last, TransOCR [10] also achieves the best performance in terms of recognition accuracy and normalized edit distance due to the strong self-attention module.

Document dataset. Since the document dataset is synthesized using the text render, the text samples in this dataset are more recognizable compared with the scene and web datasets. Besides, according to Table 3, almost no people classify the text instances as occlusion, scribble, or blur. We observe that the recognition accuracy of all baselines can exceed 90.0%. What is worth mentioning is that the CTC-based CRNN [59] achieves the best recognition accuracy in this dataset since there are no vertical text images in this dataset. In other words, the 2-D feature maps and rectification strategies are not necessary for such an easy scenario. Besides, as there are many images with long text (i.e., text length longer than 10), the attention-based recognizers (e.g., ASTER [60], MORAN [45], and SEED [54]) are subpar as they may suffer from the attention drift problem [15].

Handwriting dataset. As demonstrated in Table 3, almost 40% of the text samples in the handwriting dataset is marked as “Scribble”. Practically, the writer may join up or omit some strokes to accelerate the writing speed, which indeed brings difficulties to the existing methods. Through the experimental results, we observe that the performance varies severely across the seven baselines. For example, compared with rectification-based ASTER [60] and MORAN [45], CRNN [59] achieves better performance in this dataset as most of text samples are horizontal and with long sequences. On the contrary, the performance of semantic-based baselines SRN [88] and SEED [54] are subpar. We notice that there are many uncommon corpora in this dataset, including Tang and Song poetries, which may put obstacles to SRN and SEED to capture the semantic features. At last, TransOCR [10] achieves the best performance with the recognition accuracy of 53.4%, which further validates the strong ability of the self-attention modules.

Analysis on hard cases. We manually select some hard cases (e.g., occlusion, oblique or curved, confusing background, blur, long, and vertical) and analysis the performance of baselines on these cases. Please note that the “Scribble” situation is mainly associated with the handwriting dataset so we do not analyze the “Scribble” case separately. The experimental results on these hard cases are demonstrated in Table 6. In particular, we observe that TransOCR [10] can well adapt to each hard case compared with other baselines. Benefiting from the self-attention modules, the recognizer can flexibly associate each feature pair in the maps to enhance the sequential representation, thus making it easy to tackle uncommon text layouts like oblique and curved, or mitigate the noise induced by confusing foregrounds (occlusion) or confusing backgrounds. Interestingly, SRN [88] achieves better NED compared with TransOCR [10] in the blur situation, as it can fully exploit the semantic clues to compensate for the missing details. Additionally, the CTC-based CRNN [59] shows superiority in tackling long text referring to NED, as it can mitigate the attention drift problem incurred by the attention-based decoders. For the vertical text images, we notice that the baselines using 2-D attention maps like SAR [36], SRN [88], and TransOCR [10] performs better than the other baselines. Overall, there is still large room for improvement in terms of these hard situations.

Visualization of failure cases. We have visualized some failure cases in Figure 8, Figure 9, Figure 10, and Figure 11 for the scene, web, document, handwriting datasets, respectively. In particular, we pick ten samples from each dataset that are wrongly predicted by all the baselines. As demonstrated in Figure 8, we notice that the occluded texts (e.g., “纽慰泰”, “财经天下”, “冰淇淋鸡蛋仔”, and “海”) indeed bring difficulties to the recognizers as the frontgrounds may be mistakenly perceived as parts of the texts. Moreover, there are some extremely hard cases like mirrored texts (e.g., “麦当劳”), which are illegible even for human eyes. For the failure cases in the web dataset, we notice that the baselines are hard to tackle text images with artistic fonts (e.g., “遇见”, “魅力端午”, “没有地沟油”, and “我爱姓名贴”). As shown in Figure 10, although the document dataset is of the highest recognizability as it is synthesized by text render, some samples still pose difficulties to all baselines. We notice the baselines fail to tackle some rarely-used characters (e.g., “鼾”, “[href]”, and “瞧”), thus mistakenly regarding them as other similar characters. For the handwriting dataset, we notice that the join-up or missing strokes may confuse the baselines (e.g., “欢迎” and “火火”). Additionally, some images with long texts may make it hard for baselines to capture the sequential features.

---

9 We manually pick 100 occlusion samples, 100 oblique or curved samples, 100 confusing background samples, 100 vertical samples, 500 blur samples in the scene and web datasets. We select 1000 samples with long text (i.e., with length longer than 10) in the scene, web, document, and handwriting datasets.
6. Conclusions and Future Directions

In this article, we first discuss some reasons for the scarce attention on Chinese text recognition, e.g., lack of dataset construction standards, lack of standard evaluation procedures, lack of experimental results of the existing baselines. To tackle these problems, we collect the publicly available datasets and divide them into four categories including scene, web, document, and handwriting datasets. We then adopt seven representative methods as baselines on the collected four datasets. Through the empirical study, we observe that there are numerous challenges still under exploration. Specifically, we propose three directions for future works:

Semantic-enhanced Chinese text recognition. In Chinese, the context of phrase or character patterns can be more complicated with a huge amount of word combinations compared to English texts. Hence, it remains a challenge towards how to better learn and utilize the knowledge of the Chinese language.

Incremental learning for Chinese text recognition. As there are a huge number of Chinese characters according to Chinese national standard GB18030-2005, the recognizers may encounter zero-shot problems during the testing phase [11, 74]. In this context, incremental learning can be introduced to Chinese text recognition to alleviate the zero-shot problem by progressively expanding the size of the alphabet.

Chinese text image recovery. The text images are not always of good quality for recognition (e.g., captured with blurring, digital compression, and low-resolution appearance). Although text image recovery has been studied in the last decade, it remains an issue since the performance of current state-of-the-art methods [10, 12, 48, 72, 99] are still far from satisfactory and practical use. Besides, the existing methods mainly consider the English text images and seldom take Chinese text images into account. Under this circumstance, more research are expected in this field.

Overall, by providing the standardized dataset division and evaluation methods, we expect the benchmark can pave the way for the follow-up research.

7. Acknowledgements

We would like to thank Teng Fu, Nanxing Meng, Ke Niu and Yingjie Geng for their feedbacks on this article.

References

[1] Ouais Alsharif and Joelle Pineau. End-to-end text recognition with hybrid hmm maxout models. arXiv preprint arXiv:1310.1811, 2013. 4
[2] Jeonghun Baek, Geewook Kim, Junyop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In CVPR, 2019. 4
[3] Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa. What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In CVPR, 2021. 4
[4] Fan Bai, Zhanzhao Cheng, Yi Ni, Shiliang Pu, and Shuigeng Zhou. Edit probability for scene text recognition. In CVPR, 2018. 1
[5] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, and Yi-Zhe Song. Towards the unseen: Iterative text recognition by distilling from errors. In ICCV, 2021. 4
[6] Ayan Kumar Bhunia, Aneeshan Sain, Pinaki Nath Chowdhury, and Yi-Zhe Song. Text is text, no matter what: Unifying text recognition using knowledge distillation. In ICCV, 2021. 4
[7] Alessandro Bissacco, Mark Cummins, Yuval Netzer, and Hartmut Neven. Photocr: Reading text in uncontrolled conditions. In ICCV, 2013. 4
[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 2017. 5, 10
[9] Zhong Cao, Jiang Lu, Sen Cui, and Changshui Zhang. Zero-shot handwritten chinese character recognition with hierarchical decomposition embedding. Pattern Recognition, 2020. 5
[10] Jingye Chen, Bin Li, and Xiangyang Xue. Scene text telescope: Text-focused scene image super-resolution. In CVPR, 2021. 9, 10, 11, 12
[11] Jingye Chen, Bin Li, and Xiangyang Xue. Zero-shot chinese character recognition with stroke-level decomposition. In IJCAI, 2021. 4, 5, 12
[12] Jingye Chen, Haiyang Yu, Jianqi Ma, Bin Li, and Xiangyang Xue. Text gestalt: Stroke-aware scene text image super-resolution. In AAAI, 2022. 12
[13] Kai Chen, Li Tian, Haisong Ding, Meng Cai, Lei Sun, Sen Liang, and Qiang Huo. A compact cnn-dblstm based character model for online handwritten chinese text recognition. In ICDAR, 2017. 5
[14] Changxu Cheng, Wuheng Xu, Xiang Bai, Bin Feng, and Wenyu Liu. Maximum entropy regularization and chinese text recognition. In IJDDS, 2020. 5
[15] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In ICCV, 2017. 1, 11
[16] Zhanzhan Cheng, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Aon: Towards arbitrarily-oriented text recognition. In CVPR, 2018. 1
[17] Chee Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, et al. Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In ICDAR, 2019. 6
[18] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In ECCV, 2010. 4
[19] Jun Du, Zi-Rui Wang, Jian-Fang Zhai, and Jin-Shui Hu. Deep neural network based hidden markov model for offline handwritten chinese text recognition. In ICPR, 2016. 5
[20] 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. 5
[21] Xinjie Feng, Hongxun Yao, Yuankai Qi, Jun Zhang, and Shengping Zhang. Scene text recognition via transformer. arXiv preprint arXiv:2003.08077, 2020. 5
[22] Yunze Gao, Yingying Chen, Jinqiao Wang, Ming Tang, and Hanqing Lu. Dense chained attention network for scene text recognition. In ICIP, 2018. 4
[23] Yunze Gao, Yingying Chen, Jinqiao Wang, Ming Tang, and Hanqing Lu. Reading scene text in deep convolutional sequences. In AAAI, 2016. 1
[24] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In IML, 2006. 4, 8
[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 9
[26] Mingchao He, Yuliang Liu, Zhibo Yang, Sheng Zhang, Canjie Luo, Feiyu Gao, Qi Zheng, Yongpan Wang, Xin Zhang, and Lianwen Jin. Icpr2018 contest on robust reading for multi-type web images. In ICPR, 2018. 6
[27] Pan He, Weilin Huang, Yu Qiao, Chen Change Loy, and Xiaou Tang. Reading scene text in deep convolutional sequences. In AAAI, 2016. 4
[28] Jie Hu, Tszhang Guo, Ji Cao, and Changshui Zhang. End-to-end chinese text recognition. In GlobalSIP, 2017. 5
[29] Wenyang Hu, Xiaocong Cai, Jun Hou, Shuai Yi, and Zhiping Lin. Gtc: Guided training of ctc towards efficient and accurate scene text recognition. In AAAI, 2020. 4
[30] Yunlong Huang, Zenghui Sun, Lianwen Jin, and Canjie Luo. Epan: Effective parts attention network for scene text recognition. Neurocomputing, 2020. 4
[31] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. International journal of computer vision, 2016. 4
[32] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. Advances in neural information processing systems, 2015. 4, 8
[33] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In ICDAR, 2015. 2
[34] Dimosthenis Karatzas, Faisal Shafait, Seichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Luis Pere De Las Heras. Icdar 2013 robust reading competition. In ICDAR, 2013. 2
[35] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In CVPR, 2016. 4
[36] Minghui Liao, Jian Zhang, Zhaoyi Wan, Fengming Xie, Jiajun Liang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Scene text recognition from two-dimensional perspective. In AAAI, 2019. 5
[37] Jinghuang Lin, Zhanzhan Cheng, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Text recognition in real scenarios with a few labeled samples. In ICPR, 2021. 4
[38] 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, 2013. 5
[39] Wei Liu, ChaoFeng Chen, and Kwan-Yee K Wong. Char-net: A character-aware neural network for distorted scene text recognition. In AAAI, 2018. 4
[40] Wei Liu, ChaoFeng Chen, Kwan-Yee K Wong, Zhizhong Su, and Junyu Han. Star-net: a spatial attention residue network for scene text recognition. In BMVC, 2016. 4
[41] Xinzhao Liu, Takahito Kawanishi, Xiaomeng Wu, and Kunio Kashino. Scene text recognition with cnn classifier and wfst-based word labeling. In ECCV, 2016. 4
[72] Wenjia Wang, Enze Xie, Xuebo Liu, Wenhai Wang, Ding Liang, Chunhua Shen, and Xiang Bai. Scene text image super-resolution in the wild. In ECCV, 2020. 12
[73] Wenjia Wang, Enze Xie, Peize Sun, Wenhai Wang, Lixun Tian, Chunhua Shen, and Ping Luo. Textsr: Content-aware text super-resolution guided by recognition. arXiv preprint arXiv:1909.07113, 2019. 4
[74] Wenchao Wang, Jianshu Zhang, Jun Du, Zi-Rui Wang, and Yixing Zhu. Denseran for offline handwritten chinese character recognition. In ICFHR, 2018. 4, 5, 12
[75] Yuxin Wang, Hongtao Xie, Shancheng Fang, Jing Wang, Shenggao Zhu, and Yongdong Zhang. From two to one: A new scene text recognizer with visual language modeling network. In CVPR, 2021. 5
[76] Zi-Rui Wang and Jun Du. Writer code based adaptation of deep neural network for offline handwritten chinese text recognition. In ICFHR, 2016. 5
[77] Zi-Rui Wang and Jun Du. Joint architecture and knowledge distillation in cnn for chinese text recognition. Pattern Recognition, 2021. 5
[78] Lu Yang, Peng Wang, Hui Li, Zhen Li, and Yanning Zhang. A holistic representation guided attention network for scene text recognition. Neurocomputing, 2020. 5
[79] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained rectification network for scene text recognition. In CVPR, 2019. 4
[80] Weixin Yang, Lianwen Jin, Zecheng Xie, and Ziyong Feng. Improved deep convolutional neural network for online handwritten chinese character recognition. Pattern Recognition, 2017. 5
[81] Yao Xiao, Dan Meng, Cewu Lu, and Chi-Keung Tang. Template-instance loss for offline handwritten chinese character recognition. In ICDAR, 2019. 5
[82] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. Robustscanner: Dynamically enhancing positional clues for robust text image recognition. In ECCV, 2020. 5
[83] Fangneng Zhan and Shijian Lu. Esir: End-to-end scene text recognition via iterative image rectification. In CVPR, 2019. 4
[84] Chongsheng Zhang, Yuefeng Tao, Kai Du, Weiping Ding, Bin Wang, Ji Liu, and Wei Wang. Character-level street view text spotting based on deep multi-segmentation network for smarter autonomous driving. IEEE Transactions on Artificial Intelligence, 2021. 1
[85] Hesuo Zhang, Lingyu Liang, and Lianwen Jin. Scut-hccdoc: A new benchmark dataset of handwritten chinese text in unconstrained camera-captured documents. Pattern Recognition, 2020. 2, 6
[86] Jianshu Zhang, Yixing Zhu, Jun Du, and Lirong Dai. Ran: Radical analysis networks for zero-shot learning of chinese characters. arXiv preprint arXiv:1711.01889, 2017. 5
[87] Jianshu Zhang, Yixing Zhu, Jun Du, and Lirong Dai. Radical analysis network for zero-shot learning in printed chinese character recognition. In ICMIE, 2018. 5
[88] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Le Wang, Dong Wang, Minghui Liao, Mingkun Yang, et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In ICDAR, 2019. 2, 5, 6
[89] Xu-Yao Zhang, Yoshua Bengio, and Cheng-Lin Liu. Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark. Pattern Recognition, 2017. 5
[90] Yaping Zhang, Shuai Nie, Wenju Liu, Xing Xu, Dongxiang Zhang, and Heng Tao Shen. Sequence-to-sequence domain adaptation network for robust text image recognition. In CVPR, 2019. 4
[91] 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. In ECCV, 2020. 2, 6
[92] Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of bert for ad hoc document retrieval. arXiv preprint arXiv:1903.10972, 2019. 1
[93] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C Lee Giles. Improving offline handwritten chinese character recognition by iterative refinement. In ICDAR, 2017. 5
[94] Yao Xiao, Dan Meng, Cewu Lu, and Chi-Keung Tang. Template-instance loss for offline handwritten chinese character recognition. In ICDAR, 2019. 5
[95] Lu Yang, Peng Wang, Hui Li, Zhen Li, and Yanning Zhang. Robustscanner: Dynamically enhancing positional clues for robust text image recognition. In ECCV, 2020. 5
[96] 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. In ECCV, 2020. 2, 6
[97] Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of bert for ad hoc document retrieval. arXiv preprint arXiv:1903.10972, 2019. 1
[98] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C Lee Giles. Improving offline handwritten chinese character recognition by iterative refinement. In ICDAR, 2017. 5
[99] Yao Xiao, Dan Meng, Cewu Lu, and Chi-Keung Tang. Template-instance loss for offline handwritten chinese character recognition. In ICDAR, 2019. 5
[100] Zi-Rui Wang and Jun Du. Writer code based adaptation of deep neural network for offline handwritten chinese text recognition. In ICFHR, 2016. 5
[101] Zi-Rui Wang and Jun Du. Joint architecture and knowledge distillation in cnn for chinese text recognition. Pattern Recognition, 2021. 5
[102] Lu Yang, Peng Wang, Hui Li, Zhen Li, and Yanning Zhang. A holistic representation guided attention network for scene text recognition. Neurocomputing, 2020. 5
[103] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained rectification network for scene text recognition. In CVPR, 2019. 4
[104] Weixin Yang, Lianwen Jin, Zecheng Xie, and Ziyong Feng. Improved deep convolutional neural network for online handwritten chinese character recognition using domain-specific knowledge. In ICDAR, 2015. 5
[105] Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of bert for ad hoc document retrieval. arXiv preprint arXiv:1903.10972, 2019. 1
[106] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C Lee Giles. Improving offline handwritten chinese character recognition by iterative refinement. In ICDAR, 2017. 5
[107] Yao Xiao, Dan Meng, Cewu Lu, and Chi-Keung Tang. Template-instance loss for offline handwritten chinese character recognition. In ICDAR, 2019. 5
[108] Lu Yang, Peng Wang, Hui Li, Zhen Li, and Yanning Zhang. Robustscanner: Dynamically enhancing positional clues for robust text image recognition. In ECCV, 2020. 5
[109] 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, 2019. 6
[110] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In ECCV, 2020. 5
Figure 8. Failure cases in the scene dataset.

Figure 9. Failure cases in the web dataset.
Figure 10. Failure cases in the document dataset.

Figure 11. Failure cases in the handwriting dataset.