DTrOCR: Decoder-only Transformer for Optical Character Recognition

Masato Fujitake
FA Research, Fast Accounting Co., Ltd. Japan
fujitake@fastaccounting.co.jp

Abstract

Typical text recognition methods rely on an encoder-decoder structure, in which the encoder extracts features from an image, and the decoder produces recognized text from these features. In this study, we propose a simpler and more effective method for text recognition, known as the Decoder-only Transformer for Optical Character Recognition (DTrOCR). This method uses a decoder-only Transformer to take advantage of a generative language model that is pre-trained on a large corpus. We examined whether a generative language model that has been successful in natural language processing can also be effective for text recognition in computer vision. Our experiments demonstrated that DTrOCR outperforms current state-of-the-art methods by a large margin in the recognition of printed, handwritten, and scene text in both English and Chinese.

1. Introduction

The aim of text recognition, also known as optical character recognition (OCR), is to convert the text in images into digital text sequences. Many studies have been conducted on this technology owing to its wide range of real-world applications, including reading license plates and handwritten text, analyzing documents such as receipts and invoices [23, 58], and analyzing road signs in automated driving and natural scenes [14, 16]. However, the various fonts, lighting variations, complex backgrounds, low-quality images, occlusion, and text deformation make text recognition challenging. Numerous methods have been proposed to overcome these challenges.

Existing approaches have mainly employed an encoder-decoder architecture for robust text recognition [3, 13, 47]. In such methods, the encoder extracts the intermediate features from the image, and the decoder predicts the corresponding text sequence. Figures 1 (a)–(c) present the encoder-decoder model patterns of previous studies. The methods in Figures 1 (a) and (b) employ the convolutional neural network (CNN) [20] and Vision Transformer (ViT) [11] families as image encoding methods, with the recurrent neural network (RNN) [22] and Transformer [51] families as decoders, which can be used for batch inference (Figure 1 (a)) or recursively inferring characters one by one (Figure 1 (b)). Certain modules apply image curve correction [2] and high-resolution enhancement [38] to the input images to boost the accuracy. Owing to the limited information from images, several methods in recent years have focused on linguistic information and have employed language models (LMs) in the decoder either externally [13] or internally [3, 29], as illustrated in Figure 1 (c). Although these approaches can achieve high accuracy by leveraging linguistic information, the additional computational cost is an issue.

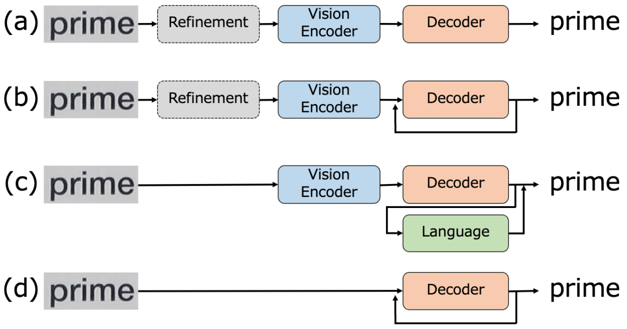

Figure 1. Model patterns for text recognition. The existing methods in (a) to (c) consist of a vision encoder to extract the image features and a decoder to predict text sequences from the features. Some methods use refinement modules to deal with low-quality images and an LM to correct the output text. Our approach differs significantly. As shown in (d), it consists of a simple model pattern in which the image is fed directly into the decoder to generate text.

LMs based on generative pre-training have been used successfully in various natural language processing (NLP) tasks [43, 44] in recent years. These models are built with a decoder-only Transformer. The model passes the input text directly to the decoder, which outputs the subsequent word token. The model is pre-trained with a large corpus to generate the following text sequence in the given text. The model needs to understand the meaning and context of the words and acquire language knowledge to generate the text sequence accurately. As the obtained linguistic information is powerful, it can be fine-tuned for various tasks in NLP. However, the applicability of such models to text recognition has yet to be demonstrated.

Motivated by the above observations, this study presents a novel text recognition framework known as the Decoder-only Transformer for Optical Character Recognition (DTrOCR), which does not require a vision encoder for feature extraction. The proposed method transforms a pre-trained generative LM with a high language representation capability into a text recognition model. Generative models that are used in NLP use a text sequence as input and generate the following text autoregressively. In this work, the model is trained to use an image as the input. The input image is converted into a patch sequence, and the recognition results are output autoregressively. The structure of the proposed method is depicted in Figure 1 (d). DTrOCR does not require vision encoders such as a CNN or ViT but has a simple model structure with only a decoder that leverages the internal generative LM. In addition, the proposed method employs fine-tuning from a pre-trained model, which reduces the computational resources. Despite its simple structure, DTrOCR is revealed to outperform existing methods in various text recognition benchmarks, such as handwritten, printed, and natural scene image text, in English and Chinese. The main contributions of this work are summarized as follows:

- We propose a novel decoder-only text recognition method known as DTrOCR, which differs from the mainstream encoder-decoder approach.
- Despite its simple structure, DTrOCR achieves state-of-the-art results on various benchmarks by leveraging the internal LM, without relying on complex pre- or post-processing.

2. Related Works

Scene Text Recognition. Scene text recognition refers to text recognition in natural scene images. Existing methods can be divided into three categories: word-based, character-based, and sequence-based approaches. Word-based approaches perform text recognition as image classification, in which each word is a direct classification class [24]. Character-based approaches perform text recognition using detection, recognition, and grouping on a character-by-character basis [53]. Sequence-based approaches deal with the task as sequence labeling and are mainly realized using encoder-decoder structures [13, 29, 47]. The encoder, which can be constructed using the CNN and ViT families, aims to extract a visual representation of a text image. The goal of the decoder is to map the representation to text with connectionist temporal classification (CTC)-based methods [18, 47] or attention mechanisms [3, 29, 32, 61]. The encoder has been improved using a neural architecture search [65] and a graph convolutional network [59], whereas the decoder has been enhanced using multistep reasoning [5], two-dimensional features [28], semantic learning [42], and feedback [4]. Furthermore, the accuracy can be improved by converting low-resolution inputs into high-resolution inputs [38], normalizing curved and irregular text images [2, 33, 48], or using diffusion models [15]. Some methods have predicted the text directly from encoders alone for computational efficiency [1, 12]. However, these approaches do not use linguistic information and face difficulties when the characters are hidden or unclear.

Therefore, methods that leverage language knowledge have been proposed to make the models more robust in recent years. ABINet uses bi-directional context via external LMs [13]. In VisionLAN, an internal LM is constructed by selectively masking the image features of individual characters during training [56]. The learning of internal LMs using permutation language modeling was proposed in PARSeq [3]. In TrOCR, LMs that are pre-trained on an NLP corpus using masked language modeling (MLM) are used as the decoder [29]. MaskOCR includes sophisticated MLM pre-training methods to enhance the Transformer-based encoder-decoder structure [34]. The outputs of these encoders are either passed directly to the decoder or intricately linked by a cross-attention mechanism.

Our approach exhibits similarities to TrOCR [29] as both use linguistic information and pre-trained LMs for the decoding process. However, our method differs in two significant aspects. First, we use generative pre-training [43] as the pre-training method in the decoder, which predicts the next word token for generating text, instead of solving masked fill-in-the-blank problems using MLM. Second, we eliminate the encoder to obtain elaborate features from images. The images are patched and directly fed into the decoder. As a result, no complicated connections such as cross-attention are required to link the encoder and decoder because the image and text information are handled at the same sequence level. This enables our text recognition model to be simple yet effective.

Handwritten Text Recognition. Handwritten text recognition (HWR) has long been studied, and the recent methods have been reviewed [35]. In addition, the effects of different attention mechanisms of encoder-decoder structures in the HWR domain have been compared [36]. The combination of RNNs and CTC decoders has been established as the primary approach in this field [6, 36, 55], with improvements such as multi-dimensional long short-term memory [41] and attention mechanisms [10, 25] having been applied. In recent years, extensions using LMs have also been implemented [29]. Thus, we tested the effectiveness of our method in HWR to confirm its scalability.

Chinese Text Recognition. Text recognition tasks on alphabets and symbols in English have been studied, and significant accuracy improvements have been achieved [3, 13]. The adaptation of recognition models to Chinese text recognition (CTR) has been investigated in recent years [34, 62]. However, studies on CTR remain lacking. CTR is a challenging task as Chinese has substantially more characters than English and contains many similar-appearing characters. Thus, we validated our method to determine whether it can be applied to Chinese in addition to English text.

Generative Pre-Trained Transformer. Generative pre-trained Transformer (GPT) has emerged in NLP, which has attracted attention owing to its ability to produce results in various tasks [43,44]. The model can acquire linguistic ability by predicting the continuation of a given text. GPT comprises a decoder-only autoregressive transformer that does not require an encoder to acquire the input text features. Whereas previous studies [29] examined the adaptation of LMs that are learned using MLM to text recognition models, this study explores the extension of GPT to text recognition.

3. Method

The pipeline of our method is illustrated in Figure 2. We use a generative LM [44] that incorporates Transformers [51] with a self-attention mechanism. Our model comprises two main components: the patch embedding and Transformer decoder. The input text image undergoes patch embedding to divide it into patch-sized image sequences. Subsequently, these sequences are passed through the decoder with the positional embedding. The decoder predicts the first-word token after receiving a special token `[SEP]` that separates the image and text sequence. Thereafter, it predicts the subsequent tokens in an autoregressive manner until the special token `[EOS]`, which indicates the end of the text, and produces the final result. The modules and training methods are described in detail in the following sections.

3.1. Patch Embedding Module

Since the input to the transformer is a sequence of tokens, the patch embedding as the input to the Transformer is a sequence of tokens, the patch embedding module transforms the tokens so that the image can be input into the decoder. We employ the patch embedding procedure proposed in [11]. The input image is resized to a fixed-size image $I \in \mathbb{R}^{W \times H \times C}$, where W, H, and C are the width, height, and channel of the image, respectively. The input image is divided by fixed patch sizes $p_w \times p_h$, where p_w and p_h are the width and height of the patch, respectively. The patch images are first transformed into vectors and adjusted to fit the input dimensions of the Transformer. Position encoding is added to preserve the information on the position of each patch. Subsequently, the resulting sequence, which contains both the transformed patches and position information, is sent to the decoder.

3.2. Decoder Module

The decoder performs text recognition using a given image sequence. The decoder initially uses the input image sequence and generates the first predicted token by following a beginning token. This token is a special token named `[SEP]`, which marks the division between the image and text sequence. Thereafter, the model uses the image and predicted token sequence to generate text autoregressively until it reaches the token `[EOS]`. The decoder output is projected by a linear layer from the dimension of the model to the vocabulary size V. Thereafter, the probabilities are computed on the vocabulary using a softmax function. Finally, the beam search is employed to obtain the final output. The cross-entropy loss function is used in this process.

The decoder uses GPT [43, 44] to recognize the text accurately using linguistic knowledge. It predicts the next word in a sentence by maximizing the entropy. Pre-trained models are publicly available, which eliminates the need for computational resources to acquire language knowledge.

The decoder comprises multiple stacks, with the Transformer layer [51] constituting one block. This block includes a multi-head mask self-attention and feed-forward network. As opposed to previous encoder-decoder structures, this method only uses a decoder for prediction, thereby eliminating the need for cross-attention between the image features and text and significantly simplifying the design.

3.3. Pre-training with Synthetic Datasets

The decoder of our method gains language knowledge through GPT in NLP. However, it does not connect this knowledge with the image information. Thus, we trained the model on artificially generated datasets that included various text forms, such as scenes, handwritten, and printed text, to aid in acquiring image and language knowledge. Further details are provided in the experimental section.

3.4. Fine-Tuning with Real-World Datasets

Recent studies have demonstrated that synthetic datasets alone are insufficient for handling real-world problems [2, 3]. Text shapes, fonts, and features may vary depending on the type of recognition required, such as printed or handwritten text. Therefore, we fine-tuned the pre-trained models using actual data for specific tasks to solve real-world text recognition issues effectively. The training procedure for the real datasets was the same as that for the synthetic ones.

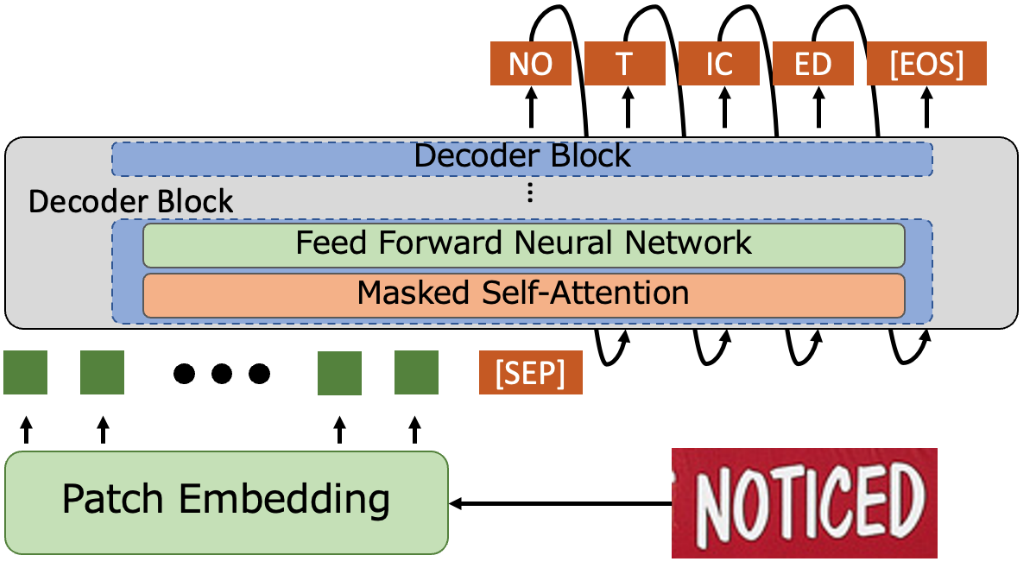

Figure 2. Architecture of proposed DTrOCR, which consists of patch embedding and decoder modules. The input images are transformed into one-dimensional sequences using the patch embedding and then sent to the decoder along with the positional encoding. The decoder uses the special token `[SEP]` to indicate sequence separation. Thereafter, it predicts the subsequent word token based on the sequence condition. It continues to generate text autoregressively until it reaches the end of the text token `[EOS]`.

3.5. Inference

The proposed method uses the same training process for inference. Patch embedding is employed for the input images and decoder to generate predicted tokens iteratively until the `[EOS]` token is reached for text recognition.

4. Experiments

			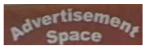
Ground truth	CHRISTMAS	PIONEER	Advertisement Space
ABINet	CHRISMMAS	LINEER	Advertisement
PARSeq	CHRUSTMAS	PIONEER	Advertisement
Ours	CHRISTMAS	PIONEER	Advertisement Space

Figure 3. Comparison of recognition results of state-of-the-art methods and proposed method [3, 13]. The result corresponding to an image is shown on each line, with the ground truth at the top. The proposed method is robust to occlusion and irregularly arranged scenes and is accurate even for two lines.

We evaluated the performance of the proposed method on scene image, printed, and handwritten text recognition in English and Chinese.

4.1. Datasets

Pre-training with Synthetic Datasets. Our proposed method was pre-trained using synthetic datasets to connect the visual and language information in the LM of the decoder. Previous studies [29] obtained training data by extracting available PDFs from the Internet and using real-world receipt data with annotations that are generated by commercial OCR. However, substantial time and effort are required to prepare such data. We created annotated datasets from a text corpus using an artificial generation method to make the process more reproducible. We used large datasets that are commonly used to train LMs as our corpus for generating synthetic text images: PILE [17] for English, CC100 [57], and the Chinese NLP Corpus[1] for Chinese with preprocessing [68].

We used three open-source libraries to create synthetic datasets from our corpus. We randomly divided the corpus into three categories: scene, printed, and handwritten text recognition, with a distribution of 60%, 20%, and 20%, respectively, to ensure text recognition accuracy. We generated four billion horizontal and two billion vertical images of text for scene text recognition using SynthTIGER [60]. We employed the default font for English and 64 commonly used fonts for Chinese. We used the Multiline text image configuration in SynthTIGER, set the word count to five, and generated 100 million images using the MJSynth [24] and SynthText [19] corpora for the recognition of multiple lines of English text.

We created two billion datasets for printed text recognition using the default settings of Text Render[2]. Addi-

[1] https://github.com/crownpku/awesome-chinese-nlp

[2] https://github.com/oh-my-ocr/text_renderer

Table 1. Word accuracy on English scene text recognition benchmark datasets with 36 characters. “Synth” and “Real” refer to synthetic and real training datasets, respectively.

Method	Training data	Test datasets and # of samples							
		IIIT5k	SVT	IC13		IC15		SVTP	CUTE
		3,000	647	857	1,015	1,811	2,077	645	288
CRNN [47]	Synth	81.8	80.1	89.4	88.4	65.3	60.4	65.9	61.5
ViTSTR$_{\text{BASE}}$ [1]	Synth	88.4	87.7	93.2	92.4	78.5	72.6	81.8	81.3
TRBA [2]	Synth	92.1	88.9	–	93.1	–	74.7	79.5	78.2
ABINet [13]	Synth	96.2	93.5	97.4	–	86.0	–	89.3	89.2
PlugNet [38]	Synth	94.4	92.3	–	95.0	–	82.2	84.3	85.0
SRN [61]	Synth	94.8	91.5	95.5	–	82.7	–	85.1	87.8
TextScanner [53]	Synth	95.7	92.7	–	94.9	–	83.5	84.8	91.6
AutoSTR [65]	Synth	94.7	90.9	–	94.2	81.8	–	81.7	–
PREN2D [59]	Synth	95.6	94.0	96.4	–	83.0	–	87.6	91.7
VisionLAN [56]	Synth	95.8	91.7	95.7	–	83.7	–	86.0	88.5
JVSR [5]	Synth	95.2	92.2	–	95.5	–	84.0	85.7	89.7
CVAE-Feed [4]	Synth	95.2	–	–	95.7	–	84.6	88.9	89.7
DiffusionSTR [15]	Synth	97.3	93.6	97.1	96.4	86.0	82.2	89.2	92.5
TrOCR$_{\text{BASE}}$ [29]	Synth	90.1	91.0	97.3	96.3	81.1	75.0	90.7	86.8
TrOCR$_{\text{LARGE}}$ [29]	Synth	91.0	93.2	98.3	97.0	84.0	78.0	91.0	89.6
PARSeq [3]	Synth	97.0	93.6	97.0	96.2	86.5	82.9	88.9	92.2
MaskOCR$_{\text{BASE}}$ [34]	Synth	95.8	94.7	98.1	–	87.3	–	89.9	89.2
MaskOCR$_{\text{LARGE}}$ [34]	Synth	96.5	94.1	97.8	–	88.7	–	90.2	92.7
SVTR$_{\text{BASE}}$ [12]	Synth	96.0	91.5	97.1	–	85.2	–	89.9	91.7
SVTR$_{\text{LARGE}}$ [12]	Synth	96.3	91.7	97.2	–	86.6	–	88.4	95.1
DTrOCR (Ours)	Synth	**98.4**	**96.9**	**98.8**	**97.8**	**92.3**	**90.4**	**95.0**	**97.6**
CRNN [3, 47]	Real	94.6	90.7	94.1	94.5	82.0	78.5	80.6	89.1
TRBA [2, 3]	Real	98.6	97.0	97.6	97.6	89.8	88.7	93.7	97.7
ABINet [3, 13]	Real	98.6	97.8	98.0	98.0	90.2	88.5	93.9	97.7
PARSeq [3]	Real	99.1	97.9	98.3	98.4	90.7	89.6	95.7	98.3
DTrOCR (ours)	Real	**99.6**	**98.9**	**99.1**	**99.4**	**93.5**	**93.2**	**98.6**	**99.1**

Table 2. Word-level recall, precision and F1 on SROIE Task 2.

Model	Recall	Precision	F1
CRNN [47]	28.71	48.58	36.09
H&H Lab [23]	96.35	96.52	96.43
MSOLab [23]	94.77	94.88	94.82
CLOVA OCR [23]	94.30	94.88	94.59
TrOCR$_{\text{LARGE}}$ [29]	96.59	96.57	96.58
DTrOCR (ours)	**98.24**	**98.51**	**98.37**

Table 3. CER on IAM Handwriting Database, where a lower score is better.

Model	Training Data	External LM	CER
Bluche *et al.* [6]	Synthetic + IAM	Yes	3.20
Michael *et al.* [36]	IAM	No	4.87
Wang *et al.* [55]	IAM	No	6.40
Kang *et al.* [25]	Synthetic + IAM	No	4.67
Diaz *et al.* [10]	Internal + IAM	Yes	2.75
TrOCR$_{\text{LARGE}}$ [29]	Synthetic + IAM	No	2.89
DTrOCR (ours)	Synthetic + IAM	No	**2.38**

tionally, we employed TRDG[3] to generate another two billion datasets for recognizing handwritten text. We followed the methods outlined in previous studies for our process [29]. We used 5,427 English and four Chinese handwriting fonts [4].

Fine-Tuning with Real-World and Evaluation Datasets. We fine-tuned the pre-trained models on each dataset and evaluated their performance on benchmarks. English scene text recognition models have traditionally been trained on

[3] https://github.com/Belval/TextRecognitionDataGenerator

[4] https://fonts.google.com/?category=Handwriting

Table 4. Word accuracy on CTR benchmark.

Method	Dataset				Parameters (M)	FPS
	Scene	Web	Document	Handwriting		
CRNN [47]	54.9	56.2	97.4	48.0	**12.4**	**751.0**
ASTER [48]	59.4	57.8	97.6	45.9	27.2	107.3
MORAN [33]	54.7	49.6	91.7	30.2	28.5	301.5
SAR [28]	53.8	50.5	96.2	31.0	27.8	93.1
SEED [42]	45.4	31.4	96.1	21.1	36.1	106.6
MASTER [32]	62.1	53.4	82.7	18.5	62.8	16.3
ABINet [13]	60.9	51.1	91.7	13.8	53.1	92.1
TrOCR [29]	67.8	62.7	97.9	51.7	83.9	164.6
MaskOCR$_{\text{BASE}}$ [34]	73.9	74.8	99.3	63.7	100	–
MaskOCR$_{\text{LARGE}}$ [34]	76.2	76.8	99.4	67.9	318	–
DTrOCR (ours)	**87.4**	**89.7**	**99.9**	**81.4**	105	97.9

large synthetic datasets owing to the limited availability of labeled real datasets. However, with the increasing amount of real-world datasets, models are now also being trained on real data. Therefore, following previous studies [2, 3], we trained our models on both synthetic and real datasets to validate the performance. Specifically, we used MJSynth [24] and SynthText [19] as synthetic datasets and COCO-Text [52], RCTW [49], Uber-Text [67], ArT [7], LSVT [50], MLT19 [39], and ReCTS [66] as real datasets. Each model was evaluated on six standard scene text datasets: ICDAR 2013 (IC13) [27], Street View Text (SVT) [54], IIIT5K-Words (IIIT5K) [37], ICDAR 2015 (IC15) [26], Street View Text-Perspective (SVTP) [40], and CUTE80 (CUTE) [45]. The initial three datasets mainly consist of standard text images, whereas the remaining datasets include images of text that are either curved or in perspective.

Thereafter, we tested the accuracy of the printed text recognition in receipt images using Scanned Receipts OCR and Information Extraction (SROIE) Task 2 [23]. A total of 626 training and 361 evaluation receipt images were used in the testing.

We employed the widely used IAM Handwriting Database to evaluate the English HWR. Aachen's partition[5] was used, which resulted in a training set of 6,161 lines from 747 forms, a validation set of 966 lines from 115 forms, and a test set of 2,915 lines from 336 forms.

We evaluated the models for CTR on a large CTR benchmark dataset [62]. This dataset includes four subsets (scene, web, document, and handwriting), with a total of 1.4 million fully labeled images. The scene subset is derived from scene text datasets such as RCTW [49], ReCTS [66], LSVT [50], ArT [7], and CTW [63]. It consists of 509,164, 63,645, and 63,646 samples for training, validation, and testing, respectively. The web subset is built on the MTWI dataset [21], with 112,471, 14,059, and 14,059 samples for training, validation, and testing, respectively. The document subset was generated in document style by Text Render and consists of 400,000 training, 50,000 validation, and 50,000 testing samples. The handwriting subset was obtained from the handwriting dataset SCUT-HCCDoc [64]. It consists of 74,603, 18,651, and 23,389 training, validation, and testing samples, respectively.

4.2. Evaluation Metrics

We used different metrics for the various benchmarks. The word accuracy was used for the standard scene text recognition and CTR benchmarks; a prediction was considered as correct if the characters at all positions matched. SROIE Task 2 was evaluated using the word-level precision, recall, and F1 scores. Finally, the character error rate (CER) with case sensitivity was used for the HWR benchmark IAM.

We followed the standard protocols for the English scene text recognition [2, 3] and CTR [62] to process the predictions and ground truth. We filtered the string to fit the 36-character character set (lowercase alphanumeric) to ensure a fair comparison in the English scene text recognition task. We implemented specific processes for the CTR: (i) full-width characters were converted into half-width ones; (ii) traditional Chinese characters were converted into simplified ones; (iii) uppercase letters were converted into lowercase ones; and (iv) all spaces were removed.

4.3. Implementation Details

We used the English[6] and Chinese[7] GPT-2 [44] models with 12 layers, 768 hidden dimensions, and a Transformer

[5] https://github.com/jpuigcerver/Laia

[6] https://huggingface.co/gpt2

[7] https://huggingface.co/uer/gpt2-chinese-cluecorpussmall

with 12 heads for our decoder model. These models use a bytepair encoding vocabulary [46], and we followed previous research [3] for image patch embedding with a size of 8×4. We used relative position encoding and set the maximum token length to 512.

We used an English pre-training dataset for English and a combination of English and Chinese datasets for Chinese to train the proposed model. Model training was performed for one epoch with a batch size of 32. We used the AdamW optimizer with a learning rate of 1e-4 [31].

During the fine-tuning phase, the models were initialized using pre-trained weights and subsequently trained for the target datasets using a learning rate of 5e-6 for one epoch, except for SROIE, for which the models were trained for four epochs. The same optimizer and batch size were used in the pre-training phase.

We approximately followed previous work for the data augmentation and label pre-processing [3]. We applied RandAugment [8], except for Sharpness. Invert, Gaussian blur, and Poisson noise were added owing to their effectiveness. The RandAugment policy of three layers and a magnitude of five was used. All images were resized to 128×32 pixels. Furthermore, the original orientation was retained, rotated clockwise, or rotated with a probability of 95%, 2.5%, and 2.5%, to account for images that were rotated 90 degrees clockwise. Eventually, the image was standardized to fit into the range of -1 to 1.

The models were trained using PyTorch on Nvidia A100 GPUs with mixed precision. The inference was performed on the RTX 2080 Ti to measure the processing speed under the same conditions as those of a previous study [62]. The reported scores are averaged from four replicates per model, following a previous study [3], except for SROIE Task 2.

4.4. Comparison with State-of-the-Art Methods

Scene Text Recognition. We compared the proposed method with several state-of-the-art scene text recognition methods. Table 1 presents the results for several widely used English benchmark datasets: IIIT5K, SVT, IC13, IC15, SVTP, and CUTE. As the training dataset conditions differed for each method, we present the results of the synthetic and real-world datasets separately. As can be observed from Table 1, our method outperformed the existing state-of-the-art methods on all benchmarks by a large margin for both the synthetic and real datasets. The competitive methods for the synthetic datasets, namely TrOCR, ABINet, PARSeq, and MaskOCR, employ an encoder and incorporate LMs to achieve high accuracy. However, our method, which does not use an encoder, achieved superior accuracy. This suggests that vision encoders are not necessary to achieve high accuracy in scene text recognition tasks. As previous studies have demonstrated that training on real datasets is more effective than that on synthetic datasets [2], we also trained our proposed method on real datasets. We confirmed that the proposed method also achieved better accuracy when it was trained on real datasets.

Figure 3 depicts the recognition results of training several state-of-the-art methods [3, 13] on real datasets. The proposed method performed text recognition reasonably well compared to the other methods, even under occlusion and an irregular layout. Moreover, the proposed method correctly read the two-line text, which methods in previous studies failed to achieve.

SROIE Task 2. Table 2 presents the results of the existing and proposed methods on SROIE Task 2. The CRNN, H&H Lab, MSOLab, and CLOVA OCR methods use CNN-based feature extractors to take advantage of image information, whereas TrOCR uses the ViT families. The results demonstrate that our method outperformed the existing methods without either approach. Thus, the proposed method can be applied not only to reading text in natural scene images but also to text in printed documents in the real world.

IAM Handwriting Database. Table 3 summarizes the results of the proposed and existing methods on the IAM Handwriting Database. Our method was superior to the most significant existing approach, which was trained with Diaz's internal annotated dataset and used an external LM [10]. Our method does not make use of either of these and can achieve better accuracy solely through synthetic and benchmark datasets. Our method also performed better than TrOCR, which uses Transformers, under similar conditions. The experimental results affirm that the proposed text recognition method with the generative LM is also effective for recognizing handwritten text.

CTR. Table 4 summarizes the results of the proposed and previous approaches on the CTR benchmark, which has more characters to categorize and is more challenging than the English benchmark. The results confirm the generality of our method. In terms of accuracy, the proposed method outperformed the existing methods by a large margin for all subsets. The Transformer-based encoder-decoder methods, namely TrOCR and MaskOCR, achieved high accuracy in previous studies. Both of those decoders are pre-trained models based on MLM. However, the proposed method is based on generative pre-training by predicting the next word token. We confirm that the decoder with the generative model can model sequential patterns more flexibly, even for complex text sequences such as Chinese.

Our method has fewer parameters and is more accurate than the existing large-scale model MaskOCR$_{\text{LARGE}}$ [34]. Therefore, our method significantly improves the tradeoff between the number of parameters and accuracy. Furthermore, the reported processing speed, which is also known as the frames per second (FPS), has been low in previous studies. Our work is based on generative LMs, and because

Table 5. Architecture analysis.

Model	Encoder	Decoder	STR	CTR
Complete model	–	12 layers (GPT-2 [44])	97.7	89.6
Model (a)	12 layers (ViT [11])	12 layers (GPT-2 [44])	97.5	90.0
TrOCR	12 layers (ViT [11])	12 layers (RoBERTa [30])	92.6	79.3

Table 6. Effects of training process.

	STR	CTR
Training from scratch	61.0	43.4
+ pre-trained decoder	88.1	81.3
+ data augmentation	95.3	88.9
+ fine-tuning with real datasets	97.7	89.6

Table 7. Effects of pre-training dataset.

Model	Dataset amount	Epochs	STR	CTR
Complete model	100 %	1	97.7	89.6
Model (b)	50 %	2	97.5	89.1
Model (c)	25 %	4	96.2	85.7
Model (d)	25 %	1	91.4	77.9

Table 8. Effects of pre-trained decoder architecture.

Model	Decoder	Parameters (M)	STR
Complete model	GPT-2	128	97.7
Model (e)	GPT-2 Medium	359	97.9
Model (f)	GPT-2 Large	778	98.3

acceleration research has recently been conducted to make LMs easier to handle [9], we believe that further improvements in terms of speed can be expected by applying these techniques.

4.5. Detailed Analysis

The English scene text recognition (STR) and Chinese Text Recognition (CTR) were used to confirm the effectiveness of the proposed method in detail. We used the real dataset for training and reported the average subset scores.

Architecture Analysis. We analyzed the effects of the model structure on the performance, as indicated in Table 5. The configurations show the number of Transformer layers and models used. Model (a) used a sequence of features that were extracted by the encoder instead of a patch-embedded image sequence. Therefore, model (a) was expected to produce more sophisticated features because it incorporates an image-specific encoder. The results show that model (a) achieved slightly higher accuracy in the CTR, whereas the proposed method was superior in the STR. As many Chinese characters appear similar in CTR, more sophisticated features may be required, resulting in a difference in the accuracy. However, sufficient accuracy was also achieved with the decoder-only structure, which indicates that an encoder is not always necessary. We also trained TrOCR, which uses RoBERTa [30] as the decoder and is pre-trained with MLM. The comparison of the decoders confirmed that the model that was pre-trained with GPT was superior in the text recognition task. Thus, the architecture analysis confirms that the text recognition model may not require the encoder-decoder architecture, and GPT is a better decoder approach.

Effects of Model Training. We verified the effects of the training process on the accuracy of our method. Table 6 presents the results for each training process. The decoder initialization using pre-trained models, data augmentation, and fine-tuning with real datasets yielded significant improvements over the training from scratch.

Effects of Pre-training Dataset. We examined the effects of the amount of pre-training datasets and training epochs, as summarized in Table 7. The proposed method was trained on the entire synthetic dataset for one epoch to avoid overfitting models. Models (b) and (c) were trained by reducing the training data using random sampling while the overall number of training iterations was maintained. The results indicate that increasing the number of data variations is more critical for our method than increasing the training iterations with fewer data. Furthermore, model (d), which was simply trained with a reduced amount of data, exhibited significantly reduced accuracy, thereby confirming the importance of the pre-training dataset size.

Model Size Effects of Pre-trained Decoder. We verified the variation in the accuracy according to the decoder. Table 8 presents the number of parameters and accuracy of each decoder. This experiment was conducted on English benchmarks owing to the limited availability of pre-trained models. The more extensive models tended to achieve higher accuracy. Thus, the experimental results confirm that a higher ability of the LM is also more critical for text recognition.

5. Conclusion

We have presented a new text recognition method known as DTrOCR, which uses a decoder-only Transformer model. In this method, a powerful generative LM that is pre-trained on a large corpus is used as a decoder, and the correspondence between the input images and recognition texts is learned. We demonstrated that in this manner, a simple text recognition structure considering linguistic information is possible. The experimental results revealed that our method outperformed existing works on various benchmarks. This study contributes to a fundamental shift in text recognition by showing that text recognition can be performed without the encoder-decoder structure approach and highlighting the possibility of decoder-only Transformer models.

References

[1] Rowel Atienza. Vision transformer for fast and efficient scene text recognition. In *ICDAR*, pages 319–334, 2021. 2, 5

[2] 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*, pages 3113–3122, 2021. 1, 2, 3, 5, 6, 7

[3] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *ECCV*, pages 178–196, 2022. 1, 2, 3, 4, 5, 6, 7

[4] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, and Yi-Zhe Song. Towards the unseen: Iterative text recognition by distilling from errors. In *ICCV*, pages 14950–14959, 2021. 2, 5

[5] Ayan Kumar Bhunia, Aneeshan Sain, Amandeep Kumar, Shuvozit Ghose, Pinaki Nath Chowdhury, and Yi-Zhe Song. Joint visual semantic reasoning: Multi-stage decoder for text recognition. In *ICCV*, pages 14940–14949, 2021. 2, 5

[6] Théodore Bluche and Ronaldo Messina. Gated convolutional recurrent neural networks for multilingual handwriting recognition. In *ICDAR*, pages 646–651, 2017. 2, 5

[7] 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*, pages 1571–1576, 2019. 6

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, pages 702–703, 2020. 7

[9] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022. 8

[10] Daniel Hernandez Diaz, Siyang Qin, Reeve Ingle, Yasuhisa Fujii, and Alessandro Bissacco. Rethinking text line recognition models. *arXiv preprint arXiv:2104.07787*, 2021. 2, 5, 7

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 3, 8

[12] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Tianlun Zheng, Chenxia Li, Yuning Du, and Yu-Gang Jiang. Svtr: Scene text recognition with a single visual model. *arXiv preprint arXiv:2205.00159*, 2022. 2, 5

[13] 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*, pages 7098–7107, 2021. 1, 2, 3, 4, 5, 6, 7

[14] Masato Fujitake. A3s: Adversarial learning of semantic representations for scene-text spotting. In *ICASSP*, pages 1–5, 2023. 1

[15] Masato Fujitake. Diffusionstr: Diffusion model for scene text recognition. *arXiv preprint arXiv:2306.16707*, 2023. 2, 5

[16] Masato Fujitake and Hongpeng Ge. Temporally-aware convolutional block attention module for video text detection. In *SMC*, pages 220–225, 2021. 1

[17] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020. 4

[18] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006. 2

[19] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016. 4, 6

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[21] Mengchao 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*, pages 7–12, 2018. 6

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *NC*, 9(8):1735–1780, 1997. 1

[23] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *ICDAR*, pages 1516–1520, 2019. 1, 5, 6

[24] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 116:1–20, 2016. 2, 4, 6

[25] Lei Kang, Pau Riba, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. Pay attention to what you read: nonrecurrent handwritten text-line recognition. *PR*, 129:108766, 2022. 2, 5

[26] 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*, pages 1156–1160, 2015. 6

[27] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *ICDAR*, pages 1484–1493, 2013. 6

[28] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, volume 33, pages 8610–8617, 2019. 2, 6

[29] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. Trocr: Transformer-based optical character recognition with pre-trained models. *arXiv preprint arXiv:2109.10282*, 2021. 1, 2, 3, 4, 5, 6

[30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 8

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, pages 1–10, 2018. 7

[32] 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. *PR*, 117:107980, 2021. 2, 6

[33] Canjie Luo, Lianwen Jin, and Zenghui Sun. Moran: A multi-object rectified attention network for scene text recognition. *PR*, 90:109–118, 2019. 2, 6

[34] Pengyuan Lyu, Chengquan Zhang, Shanshan Liu, Meina Qiao, Yangliu Xu, Liang Wu, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. Maskocr: Text recognition with masked encoder-decoder pretraining. *arXiv preprint arXiv:2206.00311*, 2022. 2, 3, 5, 6, 7

[35] Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, 8:142642–142668, 2020. 2

[36] Johannes Michael, Roger Labahn, Tobias Grüning, and Jochen Zöllner. Evaluating sequence-to-sequence models for handwritten text recognition. In *ICDAR*, pages 1286–1293, 2019. 2, 5

[37] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012. 6

[38] Yongqiang Mou, Lei Tan, Hui Yang, Jingying Chen, Leyuan Liu, Rui Yan, and Yaohong Huang. Plugnet: Degradation aware scene text recognition supervised by a pluggable super-resolution unit. In *ECCV*, pages 158–174, 2020. 1, 2, 5

[39] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *ICDAR*, pages 1582–1587, 2019. 6

[40] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *ICCV*, pages 569–576, 2013. 6

[41] Joan Puigcerver. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *ICDAR*, volume 01, pages 67–72, 2017. 2

[42] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *CVPR*, pages 13528–13537, 2020. 2, 6

[43] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 1, 2, 3

[44] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1, 3, 6, 8

[45] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *ESA*, 41(18):8027–8048, 2014. 6

[46] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, pages 1715–1725, 2016. 7

[47] 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. *TPAMI*, 39(11):2298–2304, 2016. 1, 2, 5, 6

[48] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *TPAMI*, 41(9):2035–2048, 2018. 2, 6

[49] Baoguang Shi, Cong Yao, Minghui Liao, Mingkun Yang, Pei Xu, Linyan Cui, Serge Belongie, Shijian Lu, and Xiang Bai. Icdar2017 competition on reading chinese text in the wild (rctw-17). In *ICDAR*, volume 1, pages 1429–1434, 2017. 6

[50] Yipeng Sun, Zihan Ni, Chee-Kheng Chng, Yuliang Liu, Canjie Luo, Chun Chet Ng, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, et al. Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt. In *ICDAR*, pages 1557–1562, 2019. 6

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 1, 3

[52] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016. 6

[53] Zhaoyi Wan, Minghang He, Haoran Chen, Xiang Bai, and Cong Yao. Textscanner: Reading characters in order for robust scene text recognition. In *AAAI*, volume 34, pages 12120–12127, 2020. 2, 5

[54] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV*, pages 1457–1464, 2011. 6

[55] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. Decoupled attention network for text recognition. In *AAAI*, volume 34, pages 12216–12224, 2020. 2, 5

[56] 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 *ICCV*, pages 14194–14203, 2021. 2, 5

[57] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *LREC*, pages 4003–4012, 2020. 4

[58] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *KDD*, page 1192–1200, 2020. 1

[59] Ruijie Yan, Liangrui Peng, Shanyu Xiao, and Gang Yao. Primitive representation learning for scene text recognition. In *CVPR*, pages 284–293, 2021. 2, 5

[60] Moonbin Yim, Yoonsik Kim, Han-Cheol Cho, and Sungrae Park. Synthtiger: Synthetic text image generator towards better text recognition models. In *ICDAR*, pages 109–124, 2021. 4

[61] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, pages 12113–12122, 2020. 2, 5

[62] Haiyang Yu, Jingye Chen, Bin Li, Jianqi Ma, Mengnan Guan, Xixi Xu, Xiaocong Wang, Shaobo Qu, and Xiangyang Xue. Benchmarking chinese text recognition: Datasets, baselines, and an empirical study. *arXiv preprint arXiv:2112.15093*, 2021. 3, 6, 7

[63] 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. *JCST*, 34:509–521, 2019. 6

[64] Hesuo Zhang, Lingyu Liang, and Lianwen Jin. Scut-hccdoc: A new benchmark dataset of handwritten chinese text in unconstrained camera-captured documents. *PR*, 108:107559, 2020. 6

[65] Hui Zhang, Quanming Yao, Mingkun Yang, Yongchao Xu, and Xiang Bai. Autostr: efficient backbone search for scene text recognition. In *ECCV*, pages 751–767, 2020. 2, 5

[66] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Lei Wang, Dong Wang, Minghui Liao, Mingkun Yang, et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *ICDAR*, pages 1577–1581, 2019. 6

[67] Ying Zhang, Lionel Gueguen, Ilya Zharkov, Peter Zhang, Keith Seifert, and Ben Kadlec. Uber-text: A large-scale dataset for optical character recognition from street-level imagery. In *CVPRW*, volume 2017, page 5, 2017. 6

[68] Zhe Zhao, Hui Chen, Jinbin Zhang, Xin Zhao, Tao Liu, Wei Lu, Xi Chen, Haotang Deng, Qi Ju, and Xiaoyong Du. Uer: An open-source toolkit for pre-training models. *EMNLP-IJCNLP*, page 241, 2019. 4