An Exemplar-based Framework for Chinese Text Recognition

Zhao Zhou[1,2*], **Xiangcheng Du**[1*], **Yingbin Zheng**[2†], **Xingjiao Wu**[3], **Cheng Jin**[1,4†]

[1]Shanghai Key Lab of Intell. Info. Processing, School of CS, Fudan University, Shanghai, China
[2]Videt Lab, Shanghai, China
[3]East China Normal University, Shanghai, China
[4]Innovation Center of Calligraphy and Painting Creation Technology, MCT, China
{zzhou21, xcdu22}@m.fudan.edu.cn, zyb@videt.cn, xjwu@pharm.ecnu.edu.cn, jc@fudan.edu.cn

Abstract

This paper introduces a novel exemplar-based framework for reading Chinese texts in natural scene or document images. We present the *Deep Exemplar-based Chinese Text Recognizer*, which is structured to first identify candidate characters as exemplars from each text-line, and subsequently recognize them by retrieving analogous exemplars from a database. With text-line level annotations, we design the exemplar discovery network to simultaneously recognize texts and capture individual character positions in a weak-supervision manner. The exemplar retrieval module is then crafted to identify the most similar exemplar and propagate the corresponding character label. This enables us to effectively rectify the misrecognized characters and boost the performance of scene text recognition. Experiments on four scenarios of Chinese texts demonstrate the effectiveness of our proposed framework.

Introduction

We tackle the problem of Chinese text recognition, which aims to read Chinese texts in natural scenes or documents. Along with the advances in deep convlutional networks, recent years have witnessed remarkable progress in text recognition. Many text recognition methods have made designs on the recognition of English texts, and a common belief for the state-of-the-art text recognizers is the use of sequence modeling for the text-line. While the English language represents semantic primitives with combination of letters from the Latin alphabet, Chinese is a logographic script with its characters representing a meaning or concept. Notably, there are a much larger lexicon with rich stroke patterns and many morphologically similar characters.

In this paper, we present a novel text representation and training framework based solely on character-level information we call *Deep Exemplar-based Chinese Text Recognizer* (*DECTR*). Instead of the sequence modeling such as the recurrent network to handle text sequences (Shi, Bai, and Yao 2017; Fang et al. 2021), we are inspired by the pipeline of image retrieval (Noh et al. 2017; Radenović, Tolias, and Chum 2018; Weinzaepfel et al. 2022), and the framework is built upon candidate Chinese character exemplars being selected and recognized along with our algorithmic pipeline. Figure 1 illustrates our approach. Starting from training image set with text-line level annotations, an *exemplar discovery network* is designed to simultaneously recognize texts and estimate individual character positions in a weak-supervision manner. With an image going through this network, we define its exemplars as the output positions and regional features of potential characters; the corresponding character label prediction is considered as a pseudo label for the exemplar. The exemplar bank is composed of character exemplars and labels extracted from the training data or text-line images from other sources. We further introduce the *exemplar retrieval module*, where the exemplars extracted from the testing image are sent to retrieve in the exemplar bank and the label of the matched exemplar is propagated.

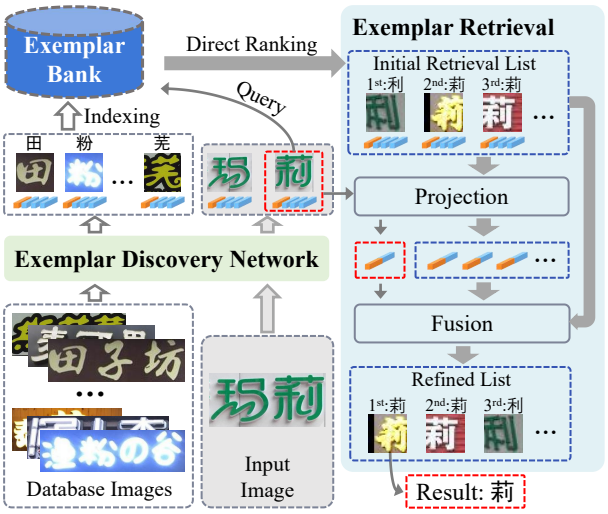

Figure 1: DECTR pipeline. We take the second exemplar of the input image to show the steps during exemplar retrieval, and the same process will also be applied to other exemplars.

The highlight of the proposed framework is threefold. First, the framework fully utilizes character-level information and variations for the Chinese texts. Our recognition system can boost performance by adding items into the exemplar bank under the exemplar discovery-retrieval

*These authors contributed equally.
†Corresponding author.

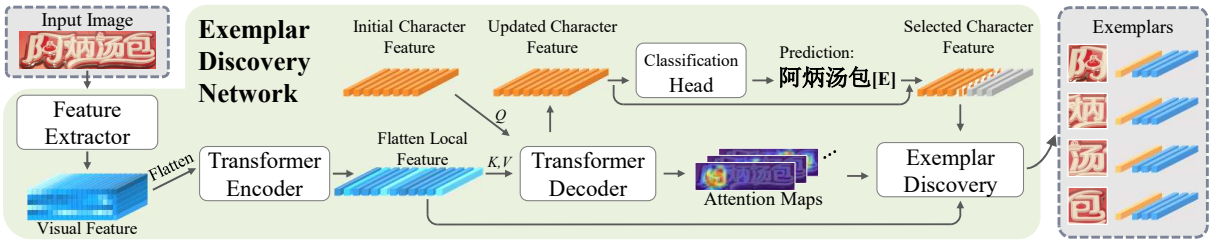

Figure 2: Architecture of Exemplar Discovery Network. [E] indicates the end-of-sequence character.

pipeline, rather than fine-tuning the whole network that demands more computational workloads. Second, we set up a compact and effective structure for the Exemplar Discovery Network. Concatenating the predicted exemplar labels produces the image-level recognition results. Intriguingly, we observe that these results are on par with the performance of state-of-the-art sequence-based methods, even those employing more complex networks. This confirms the significance of incorporating character-level configurations into the model. Third, compared with directly using visual representations of the exemplars during retrieval, a projection network is used to transform them into a new feature space for retrieval. The structure is lightweight and mainly utilizes inter-character diversity. This is especially useful for Chinese characters with large variations, and we demonstrate a considerable improvement in this strategy. We also perform our framework on the Chinese text recognition benchmark with four types of real-world Chinese texts, and the extensive experiments demonstrate the superior performance of the proposed DECTR compared with previous approaches.

Related Work

Many approaches treat the text recognition task as a sequence prediction problem, by a two-stage approach for feature extraction and text transcription (Shi, Bai, and Yao 2017; Shi et al. 2018; Baek et al. 2019; Du et al. 2020). There are also many works that incorporate Semantic context into the recognition models, such as edit probability from lexicon (Zhan and Lu 2019) and linguistic rules (Fang et al. 2021). However, due to the complexity and diversity of the visual appearance for Chinese characters, here we focus on the harness of the visual representations to effectively recognize texts and our work is under a lexicon-free setting.

The proposed method is closely related to the character-level text recognizers. Classic character-level methods (Sheshadri and Divvala 2012; Yao et al. 2014) focus on describing the characters, while recent methods aim to make bottom-up text recognition with discriminative deep features of the characters, such as part prototypes (Cao et al. 2020). (Peng et al. 2022; Yu et al. 2024) apply the weakly supervised learning of the character instances for handwritten Chinese text recognition. Recently, several frameworks utilize matching-based techniques with deep features to recognize the characters (Zhang, Gupta, and Zisserman 2020; Liu, Yang, and Yin 2022; Souibgui et al. 2022). Among them, (Zhang, Gupta, and Zisserman 2020) turns text recognition into visual matching by constructing a glyph exemplar set and achieving promising results in documents. Compared with these previous approach, our DECTR framework builds the exemplar bank with weak supervision and the utilization of exemplar discovery-retrieval pipeline is proved to be suitable for Chinese text recognition.

There are several key components in our framework. The transformer structure is popular in recent scene text recognition approaches. For example, the vision model of (Fang et al. 2021) employs Transformer units for feature extraction and sequence modeling. SVTR (Du et al. 2022) decomposes an image into small patches and carries out hierarchical stages with Transformer. Our framework is also closely related to scene text retrieval (Wen et al. 2023). (Wen et al. 2023) treats similarity matching in text-line level, while ours focus on the character level to the combinatorial explosion with a large lexicon for Chinese. The exemplar retrieval module in our framework is inspired by (Noh et al. 2017), in which an attention mechanism is designed to identify semantically useful local features. In this work, we try to leverage the mechanism in the Chinese character regions. We also extend with the fusion of retrieval results in each step and finally achieve outstanding results.

Framework

We now elaborate the construction of the proposed deep exemplar-based framework. We start by introducing the exemplar discovery network to discover exemplars from images, followed by the exemplar retrieval module to propagate the label of retrieved exemplar.

Exemplar Discovery Network

Constructing a robust visual representation is important to the success of a text recognition system. Ideally, we want compact and discriminant features so that each Chinese character in text-line can be separated and classified. With these in mind, we design this module with the transformer-based architecture with impressive performance, and the structure of the proposed exemplar discovery network is shown in Figure 2. The entire image is initially fed to the feature extractor which consists of several convolutional layers to generate the initial visual features. We aim to model the image sequence and locate the potential character regions simultaneously. Therefore, instead of the encoder-only structure like ViT (Dosovitskiy et al. 2021), we employ an encoder-decoder structure (Carion et al. 2020). The encoder takes the flattened visual features as input and enhances them with several transformer units. A fixed num-

ber of learned positional embeddings, namely the character features in Figure 2, are taken as input of the transformer decoder and attend to the encoder outputs. A prediction is made with a classification head and updates the character features. Finally, the selected character features, as well as the attention maps from the decoder, are sent to the exemplar discovery component and generate the exemplars. Compared with previous methods that employ sliding windows based text segmentation (Zhang, Gupta, and Zisserman 2020), the learned exemplars has stronger adaptability to the affine transformations in natural scenes, especially for the Chinese character with various widths. We describe below each individual components in detail.

Feauture Extractor Rich feature representations are produced. We leverage the residual blocks (He et al. 2016) that consist of a few convolutional and normalization layers. This is a common practice among many state-of-the-art transformer-based frameworks (Cheng et al. 2022) to ensure that text recognition can benefit from the training datasets. Suppose the input of this subnet is an image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$, where H and W represent the height and width of the image, respectively. We make a trade-off between speed and precision and design a compact structure for feature extraction[1]. The output feature maps are considered as the visual features $\boldsymbol{v} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$ for the subsequent modules.

Transformer Encoder As shown in Figure 2, the transformer encoder takes the feature maps from the previous feature extractor as input. We expect the encoder can capture dependencies between all positions in the visual feature $\boldsymbol{f}$, as the spatial relationships between different parts of the image may provide valuable information for recognizing the Chinese characters. We employ the design from (Zhu et al. 2020), which provides an efficient deformable attention mechanism. A block of transformer encoder is composed of a multi-head deformable-attention (MDA) and a fully connected feed-forward network (FFN). Formally, let $\boldsymbol{v}_f \in \mathbb{R}^{\frac{HW}{16}\times C}$ be the flatten of the visual feature $\boldsymbol{v}$, $\boldsymbol{z}_q \in \mathbb{R}^{\frac{HW}{16}\times C}$ be the query feature derived by adding $\boldsymbol{v}_f$ and the position embedding, and $\boldsymbol{p}_q \in \mathbb{R}^2$ be the corresponding reference point. The multi-head attention is computed by $\mathtt{MDA}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{v}_f) = \sum_i \boldsymbol{W}_i \mathtt{DA}_i(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{v}_f)$, where $\mathtt{DA}_i(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{v}_f)$ indicates the deformable attention as described in (Dai et al. 2017).

The update of an encoder layer can be expressed as:

$$\boldsymbol{z}'_l = \mathtt{LN}(\mathtt{MDA}(\boldsymbol{z}_{l-1}, \boldsymbol{p}, \boldsymbol{v}) + \boldsymbol{v}) \tag{1}$$

$$\boldsymbol{z}_l = \mathtt{LN}(\mathtt{FFN}(\boldsymbol{z}'_l) + \boldsymbol{v}) \tag{2}$$

where LN represents layer normalization, FFN is a feed-forward network, and l is layer index. The transformer encoder consists of L_{enc} encoder layers and we will examine the influence of L_{enc} in the ablation study. The features from the last encoder layer $\boldsymbol{z}_{\text{enc}}$ are considered as the local representation for the subsequent steps.

[1]The structure of the feature extractor as well as its performance are detailed in the supplementary material.

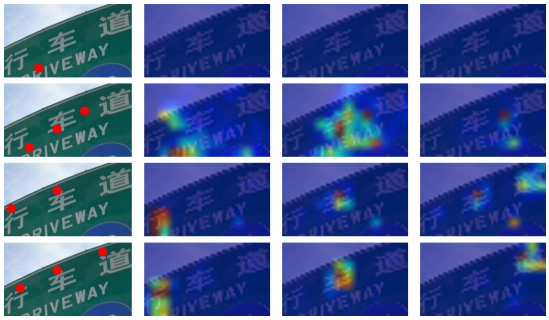

Figure 3: The attention maps corresponding to the first three character features. Row 1: initial attention maps; Row 2-4: attention maps after 1st, 4th, 16th epoch training.

Transformer Decoder The decoder follows the main steps of DETR (Carion et al. 2020). A fixed number of text query embeddings are added as input of the attention layer. We design these embeddings to represent the learned potential character-level positional encodings and update them through the decoder process, and we refer to them as character features. For each decoder block, the input is N character features $\boldsymbol{x} \in \mathbb{R}^{N\times C}$ as well as the encoder output $\boldsymbol{z}_{\text{enc}}$, and the output is the updated character features as follow,

$$\boldsymbol{x}'_l = \boldsymbol{A}_l \boldsymbol{V}^z + \boldsymbol{x}_{l-1} \tag{3}$$

$$\boldsymbol{x}''_l = \mathtt{MSA}(\mathtt{LN}(\boldsymbol{x}'_l)) + \boldsymbol{x}'_l \tag{4}$$

$$\boldsymbol{x}_l = \mathtt{LN}(\mathtt{FFN}(\boldsymbol{x}''_l) + \boldsymbol{x}_{l-1}) \tag{5}$$

Here MSA is the standard multi-head self-attention operation, l is layer index, $\boldsymbol{A}_l = \mathtt{softmax}((\boldsymbol{Q}^x_{l-1})(\boldsymbol{K}^z)^T/\sqrt{C})$ is the attention maps within the cross-attention computing, $\boldsymbol{Q}^x_{l-1} \in \mathbb{R}^{N\times C}$, $\boldsymbol{K}^z, \boldsymbol{V}^z \in \mathbb{R}^{\frac{HW}{16}\times C}$ stand for the linearly projected features for query, key, value from $\boldsymbol{x}_{l-1}$, $\boldsymbol{z}_{\text{enc}}$, $\boldsymbol{z}_{\text{enc}}$. Different from the standard DETR, we first employ the cross-attention on the character features and the output from the encoder, then apply self-attention. This is inspired by (Cheng et al. 2022) for computation more effective.

There are two goals for the design of the transformer decoder. The first purpose is to have the positions of the potential exemplar characters. Most existing scene text recognition datasets are labeled at the text-line level, and we do not employ extra annotation for the character. With the proposed attention-based decoder, the character regions can be learned in a weak-supervision manner. Figure 3 depicts the changes in the attention maps during training. The peak values in the attention maps (red points) can be observed significantly after a few epochs. With the supervision of text-line character sequencing in the follow-up classification head, these attended regions are usually ordered, i.e., the attended region in the second attention map corresponding to the second character in the annotation. This ensures the detection of the character sequences. We also obtain the character features for the exemplars, which will be used in the classification head as well as in the exemplar retrieval module.

Classification Head and Exemplar Discovery For the character features output from the last decoder block, we

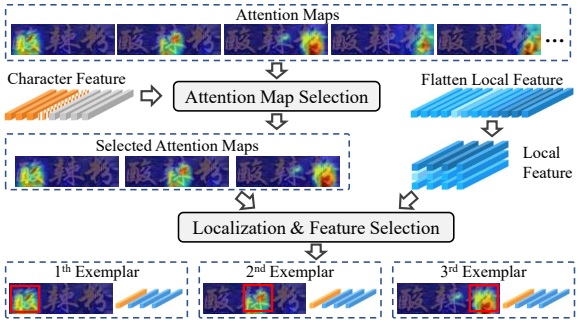

Figure 4: The data flow of character exemplar discovery.

subsequently use a classification head that contains an fully-connected layer. The final step in this network is the discovery of exemplars, as illustrated in Figure 4. The attention maps are first selected depending on whether the character features are omitted. The flattened local feature output from the transformer encoder is another input for this module. It is reshaped with aspect ratio of the original image. For a selected attention map $\boldsymbol{A}$, the position of the peak value is identified. A square region with a size of $\frac{H}{4} \times \frac{H}{4}$ is located, and the local features of the points with the top-n high attended values $(\boldsymbol{p}_1, ..., \boldsymbol{p}_n)$ are considered as the local features for this character exemplar. Given a text-line image, the overall output for the exemplar discovery component (as for the exemplar discovery network) is an exemplar set $\{\boldsymbol{E}_1, \boldsymbol{E}_2, ..., \boldsymbol{E}_N\}$ where the tuple $\boldsymbol{E}_i = (\boldsymbol{x}_i, \boldsymbol{Z}_i, y_i)$ consists of character features $\boldsymbol{x}_i$, set of local features $\boldsymbol{Z}_i = \{\boldsymbol{z}_{\boldsymbol{p}_1}, ..., \boldsymbol{z}_{\boldsymbol{p}_n}\}$, and character prediction y_i.

Exemplar Retrieval

Chinese exhibits more diversity of categories than English: its lexicon contains thousands of characters, and many of them have morphological similarities to others. The exemplar retrieval aim to prevent some misclassification from the classification head with retrieval of matched exemplars with the same category. The performance may be further boosted by incorporating more exemplars with different fonts or styles. The pseudo-code and steps of the exemplar retrieval module are listed in Algorithm 1.

Exemplar Bank and Direct Ranking During the construction of the exemplar bank, we consider the "pseudo" labels for the exemplars, i.e., the corresponding character label prediction by the exemplar discovery network. The pseudo labels are not the ground-truth labels and there exists some mistakes caused by bias of position localization or misclassification. We use two strategies to strengthen the exemplar bank. First, we filter the candidates from the text-line images with a different length between ground-truth and prediction texts. Second, we consider the confidence score s for the character prediction, since a lower score may lead to error classification; only the exemplars with a score higher than a threshold are sent to the exemplar bank.

A straightforward approach is to directly retrieve with the feature representing the whole region of character, i.e., the character feature $\boldsymbol{x}$; we denote this procedure as "direct ranking". We compute the distance by $\texttt{dist}_g(\boldsymbol{E}_i, \boldsymbol{E}_j) = L_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The top-$k$ exemplars with low distance value with will be collected into the initial list.

Algorithm 1: Exemplar Retrieval

Input: Query exemplar $\boldsymbol{E}_q$; exemplar bank $\{\boldsymbol{E}\}$.
Output: Character prediction y'_q for $\boldsymbol{E}_q$.

$\boldsymbol{x}_* \leftarrow$ character feature of $\boldsymbol{E}_*$
$\boldsymbol{Z}_* \leftarrow$ local feature set of $\boldsymbol{E}_*$
for each $\boldsymbol{E}$ in $\{\boldsymbol{E}\}$ **do**
    Compute distance: $d^g \leftarrow \texttt{dist}_g(\boldsymbol{E}_q, \boldsymbol{E})$.
end for
Obtain initial list $\{\boldsymbol{E}_1,...,\boldsymbol{E}_k\}$ with distance $\{d^g_1,...,d^g_k\}$.
Project query feature: $\boldsymbol{x}'_q \leftarrow \Phi(\boldsymbol{x}_q, \boldsymbol{Z}_q)$.
for $i = 1$ to k **do**
    Project feature for candidate: $\boldsymbol{x}'_i \leftarrow \Phi(\boldsymbol{x}_i, \boldsymbol{Z}_i)$.
    Compute distance: $d^p_i \leftarrow \texttt{Dist}_p(\boldsymbol{E}_q, \boldsymbol{E}_i)$.
    Fuse distance: $d_i \leftarrow \texttt{Fusion}(d^g_i, d^p_i)$.
end for
Find nearest distance: $d_j \leftarrow \texttt{argmin}(d_1, ..., d_k)$.
Assign prediction: $y'_q \leftarrow y_j$.
return y'_q.

Projection The direct ranking focuses on the character-level representation. For recognizing the morphologically similar characters, local regions are also important, as the difference may occur in a small portion. Therefore, we propose a very lightweight and efficient projection network to integrate the character feature $\boldsymbol{x}$ and local feature set $\boldsymbol{Z} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_n\}$. Formally, the network is defined as

$$\texttt{Pool}(\boldsymbol{Z}) = \texttt{Pool}(\boldsymbol{z}_1, ..., \boldsymbol{z}_n) \tag{6}$$

$$\hat{\boldsymbol{x}} = \texttt{Concat}(\boldsymbol{x}, \texttt{Pool}(\boldsymbol{Z})) \tag{7}$$

$$\Phi(\boldsymbol{x}, \boldsymbol{Z}) = \texttt{Linear}(\texttt{LN}(\texttt{FFN}(\hat{\boldsymbol{x}})) + \hat{\boldsymbol{x}}) \tag{8}$$

A pooling operation $\texttt{Pool}$ is first applied to the local features. We evaluate the different pooling methods, including average pooling, max pooling, and generalized mean pooling (Radenović, Tolias, and Chum 2018), and find that average pooling is more suitable for our framework. After concatenating the character feature and pooled local feature, we employ a simple feed-forward network and then project into a new feature $\boldsymbol{x}' \in \mathbb{R}^{C'}$. The projection network is very compact, with 2.1M parameters and a computational cost of 0.004 GFLOPs. The distance of projected features is measured in the same manner as for the character features, i.e., $\texttt{dist}_p(\boldsymbol{E}_i, \boldsymbol{E}_j) = L_2(\boldsymbol{x}'_i, \boldsymbol{x}'_j)$. The experiments show that the projected features offer a refined representation of characters, and heighten the ability to distinguish characters that are morphologically similar yet unique characteristics, thus improving the performance of the DECTR framework.

Training of the projection network is based on the data from the exemplar bank and we employ the Triplet loss (Chechik et al. 2010) for the model learning. Specifically, we randomly sample an exemplar $\boldsymbol{E}_i$ from the bank and query in the bank with the direct ranking function. The exemplar retrieval system returns a set of exemplars. Among

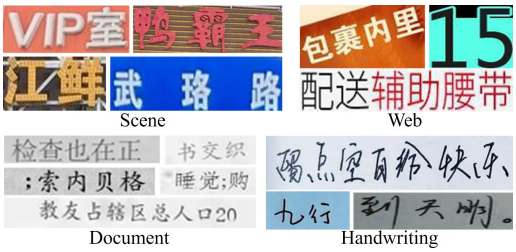

Figure 5: Sample images from (Yu et al. 2021).

them, the samples with the identical character prediction label are considered positive samples, while other samples are negative. We then sample one positive exemplar $\boldsymbol{E}_{ij}^{p}$ and one negative exemplar $\boldsymbol{E}_{ij}^{n}$ and form the triplet $\{\boldsymbol{E}_i, \boldsymbol{E}_{ij}^{p}, \boldsymbol{E}_{ij}^{n}\}$. The marginal loss function for projection network $\mathcal{L}_{\text{pn}}$ is defined as $\mathcal{L}_{\text{pn}} = \sum_i \sum_j \max\left\{d_{ij}^{p} - d_{ij}^{n} + \xi, 0\right\}$, where $d_{ij}^{p} = \texttt{dist}_p(\boldsymbol{E}_i, \boldsymbol{E}_{ij}^{p})$ and $d_{ij}^{n} = \texttt{dist}_p(\boldsymbol{E}_i, \boldsymbol{E}_{ij}^{n})$ are the distance between exemplars, ξ denotes a margin.

Fusion The final step is the fusion of the features before and after the projection network. We adopt the late fusion strategy by considering distance over different feature types:

$$\texttt{Fusion}(d^g, d^p) = \lambda_f d^g + (1 - \lambda_f) d^p \tag{9}$$

Where λ_r is a trade-off parameter between different stages and we set it as 0.7 by experiment on the validation set. The exemplars are reranked according to the fused distance, and the character label of the top-ranking exemplar is propagated to the query exemplar as its character prediction.

Experiments

Setup We use the datasets from the Chinese text recognition benchmark (Yu et al. 2021) and follow the standard protocols. Images with four types of real-world Chinese texts are used for evaluation, i.e., scene, web, document, and handwriting texts (Figure 5). Each dataset presents unique challenges, which allow us to thoroughly assess the performance. The scene/web/document datasets contain 636,455/140,589/500,000 text-line images in total, with a proportion of 8:1:1 for training, validation, and testing. The handwriting dataset contains 74,603 samples for training, 18,651 for validation and 23,389 for testing.

To evaluate the recognition performance, we use both word-level and character-level results following (Fang et al. 2021; Yu et al. 2021). For word-level evaluation, the accuracy over the whole testing set (ACC) is employed. The character-level results are measured by the average of the normalized edit distance (NED) between the predicted texts and ground-truth texts. Additionally, we measure the computational cost of our model using giga floating point operations (GFLOPs), which is a hardware-independent metric.

Implementation Details The model training in our framework comprises two phases. We first train the exemplar discovery network to obtain the initial recognition results, along with the location and features of potential characters.

	Input size	Feature Extractor	L_{enc}	L_{dec}
EDN-S	32×256	[1,1,1,1,0], 256	3	2
EDN-B	32×256	[1,1,1,1,1], 512	3	3
EDN-L	64×512	[1,1,1,1,1], 512	6	3

For feature extractor, "$[L_1, L_2, L_3, L_4, L_5], C$" corresponds to the layer numbers (L_1 to L_5) and the output channels C. L_{enc} and L_{dec} are the number of the encoder/decoder layers.

Table 1: Configurations for variants of EDN.

Subsequently, with the representations of the character exemplars, we train the network in exemplar retrieval module.

During training, we do not employ the rectification module to normalize the input text image. Our framework is implemented using PyTorch (Paszke et al. 2019). The networks are trained from scratch, using the Adam optimizer (Kingma and Ba 2015) with initial learning rate 10^{-4}. The models are trained with batch size of 80 for 40 epochs in total. We use 4 NVIDIA Titan Xp GPUs to train the networks and the inference is conducted in a single GPU.

Ablation Study

We first provide ablation studies and discussion to justify design choices of exemplar discovery network (EDN for short), which is the basic of the whole framework. Three variants are designed with different specifications and capacities, i.e., EDN-S (Small), EDN-B (Base), and EDN-L (Large), as listed in Table 1. Without further statements, we employ the structure and configuration of EDN-B as the default settings. The results for EDN are measured based on the prediction after classification head.

For the exemplar retrieval module, we start by exploring the hyper-parameters for direct ranking, projection, and fusion. Finally, an experiment with synthetic data is designed to validate the scalability of the proposed method.

Encoder and Decoder in EDN We first evaluate the effect of encoder and decoder layers. Table 2(a) shows the performance of our network with different layer numbers of transformer encoders. $L_{\text{enc}} = 0$ indicates that we omit the encoder structure and directly send the representations from feature extractor to decoder. We observe significant performance gains when the number of encoder layers increases from zero to six, while the parameters and GFLOPs of the encoder grow linearly with the layer number. Using more layers does not further improve the results.

In Table 2(b), we list the results versus the number of decoders. Using only 2 decoder layers, we can already get an accuracy of 68.91% and NED of 0.845. Adding more filters further boosts the accuracy until 4 layers, after which the performance tends to be saturated.

Transformer-based Text Recognizers We compare EDN with previous Transformer-based methods. The first is the vision model of ABINet (ABINet-VM, (Fang et al. 2021)), by employing ResNet and transformer units. SVTR (Du et al. 2022) is a transformer-based method with patch-wise image tokenization. We also compare with MaskOCR-ViT (Lyu et al. 2022) that employs ViT with masked encoder-decoder pretraining.

L_{enc}	ACC	NED	#Params	GFLOPs
0	59.39%	0.787	11.26M	2.54
3	69.47%	0.848	19.28M	6.65
6	71.61%	0.861	27.31M	10.76
9	67.13%	0.824	35.34M	14.87

(a) Different number of encoders.

L_{dec}	ACC	NED	#Params	GFLOPs
1	67.57%	0.837	17.18M	6.54
2	68.91%	0.845	18.23M	6.60
3	69.47%	0.848	19.28M	6.65
4	69.65%	0.851	20.34M	6.70
5	69.13%	0.839	21.39M	6.76
6	67.63%	0.829	22.44M	6.81

(b) Different number of decoders.

Table 2: Ablation on encoder and decoder.

Methods	Accuracy	#Params
ABINet-VM (Fang et al. 2021)	66.7%	42.5M
SVTR-T (Du et al. 2022)	67.9%	6.0M
SVTR-S (Du et al. 2022)	69.0%	10.3M
SVTR-B (Du et al. 2022)	71.4%	24.6M
SVTR-L (Du et al. 2022)	72.1%	40.8M
MaskOCR-ViT-S (Lyu et al. 2022)	68.6%	36.0M
MaskOCR-ViT-B (Lyu et al. 2022)	68.8%	100M
MaskOCR-ViT-S†(Lyu et al. 2022)	71.4%	36.0M
MaskOCR-ViT-B†(Lyu et al. 2022)	73.9%	100M
EDN-S	65.56%	7.72M
EDN-B	69.47%	19.3M
EDN-L	73.94%	27.3M

Results of SVTR and MaskOCR-ViT are reported in the original paper; ABINet-VM result is from our re-implementation; † indicates the network is with vision-language pretraining.

Table 3: Results of the transformer based methods.

Table 3 summarizes the accuracies and parameter sizes of different approaches in the Chinese scene text dataset. From the table, we see that EDN outperforms the previous methods with a similar parameter size, and achieve comparable result with MaskOCR-ViT-B† that employ pretraining and large network structure. Moreover, EDN can return the localization of the character exemplars and the performance can be further boosted with the exemplar retrieval module.

Classification vs. Retrieval As mentioned in previous section, the direct ranking is a straightforward approach that retrieves the query character features with the features from the exemplar bank directly. The label of the first retrieved exemplar in initial retrieval list is considered as the prediction of query exemplar, and the recognition result of a textline is the concatenation of predictions from its exemplars. In Table 4, we list results of the four Chinese text datasets with the exemplars extracted with an EDN-B. For all the data types, the ranking-based method (DR, Line 2) outperforms the classification-based method (CB, Line 1).

We also design a simple combination of these two methods, by first thresholding the classification scores and only sending exemplars with the score lower than threshold (0.7) into direct ranking. We denote this strategy as direct ranking with threshold (DR-T). As shown in Table 4, DR-T outperforms both classification-based method and direct ranking, indicating the two methods can be complementary. We employ this direct ranking with threshold strategy for the following experiments.

Methods	Scene	Web	Document	Handwriting
CB	69.47%	61.90%	98.05%	48.98%
DR	71.23%	62.48%	98.24%	49.21%
DR-T	72.32%	64.18%	98.47%	51.90%

Table 4: Effect of direct ranking on different datasets.

#Exemplar	ACC	NED	#Exemplar	ACC	NED
Top 1	72.32%	0.855	Top 10	73.10%	0.865
Top 2	72.86%	0.862	Top 20	73.11%	0.865
Top 5	73.07%	0.864	Top 30	73.10%	0.865

(a) Different number of exemplars used during training.

Strategy	ACC	NED	GFLOPs
RRT	73.16%	0.865	0.225
Ours	73.10%	0.865	0.014

(b) Feature combination strategy.

Table 5: Ablation on the projection network.

Projection and Fusion for Retrieval Recall that we train the projection network with Triplet loss by constructing positive and negative retrieved exemplars; here we also examine the number of used exemplars. Table 5(a) list the results versus the number of exemplars per query. We observe that using 10 exemplars makes a trade-off between performance and the triplet number. The whole exemplar retrieval module is also related to the re-ranking procedure for the image retrieval task. Therefore, we compare our method with RRT (Tan, Yuan, and Ordonez 2021), a state-of-the-art transformer-based re-ranking approach. The performance of both methods is similar, while our method requires a much smaller computational cost (Table 5(b)).

We also evaluate the importance of the exemplar retrieval module with variants of EDN in four text recognition scenarios. We denote our framework that employs the exemplar discovery network variants EDN-S, EDN-B, and EDN-L as DECTR-S, DECTR-B, and DECTR-L, respectively. As listed in the bottom of Table 6, significant improvements are achieved in the scene, web, and handwriting datasets. The changes on the document images are minor, as the results on EDNs are relatively high. Another observation is that the performance of the exemplar retrieval with a smaller exemplar discovery network is on par with directly using a larger network (e.g., DECTR-S *vs.* EDN-B).

Scalability To further validate whether the performance of exemplar retrieval is affected by the exemplar bank, we add another simple experiment to evaluate the scalability of the module. We employ the EDN-B trained from the web training set and evaluate the performance on a subset of the web testing images that only contains the Chinese characters. With this protocol, the accuracy from the classification head of EDN-B is 68.01% (the dashed line), and the exemplar retrieval with exemplar bank constructed from the training images reaches an accuracy of 68.62% (the red asterisk).

Approaches	Scene	Web	Document	Handwriting	#Params
CRNN (Shi, Bai, and Yao 2017)	54.94% / 0.742	56.21% / 0.745	97.41% / 0.995	48.04% / 0.843	**12.4M**
ASTER (Shi et al. 2018)	59.37% / 0.801	57.83% / 0.782	97.59% / 0.995	45.90% / 0.819	27.2M
MORAN (Luo, Jin, and Sun 2019)	54.68% / 0.710	49.64% / 0.679	91.66% / 0.984	30.24% / 0.651	28.5M
TransOCR (Chen, Li, and Xue 2021)	67.81% / 0.817	62.74% / 0.782	97.86% / 0.996	51.67% / 0.835	83.9M
ABINet (Fang et al. 2021)	60.88% / 0.775	51.07% / 0.704	91.67% / 0.987	13.83% / 0.514	53.1M
SVTR-L (Du et al. 2022)	72.1% / -	- / -	- / -	- / -	40.8M
MaskOCR-ViT-B (Lyu et al. 2022)	68.8% / -	70.7% / -	98.6% / -	49.4% / -	100.0M
MaskOCR-ViT-B† (Lyu et al. 2022)	**73.9%** / -	**74.8%** / -	**99.3%** / -	**63.7%** / -	100.0M
CCR-CLIP (Yu et al. 2023)	71.31% / **0.829**	69.21% / **0.797**	98.29% / **0.997**	60.30% / **0.849**	184.0M
EDN-S	65.56% / 0.824	60.87% / 0.783	98.30% / 0.996	43.39% / 0.802	**7.7M**
EDN-B	69.47% / 0.848	62.62% / 0.797	98.31% / 0.996	49.02% / 0.842	19.3M
EDN-L	73.94% / 0.877	67.49% / 0.833	98.36% / 0.997	55.21% / 0.880	27.3M
DECTR-S	69.44% / 0.842	62.15% / 0.784	98.55% / 0.996	46.84% / 0.809	12.4M
DECTR-B	73.10% / 0.865	65.62% / 0.811	98.52% / 0.996	52.87% / 0.857	24.0M
DECTR-L	**77.44%** / **0.893**	**70.02%** / **0.844**	**99.20%** / **0.998**	**59.03%** / **0.894**	32.0M

Table 6: Comparison with state-of-the-arts. The recognition performances are measured in ACC/NED, respectively.

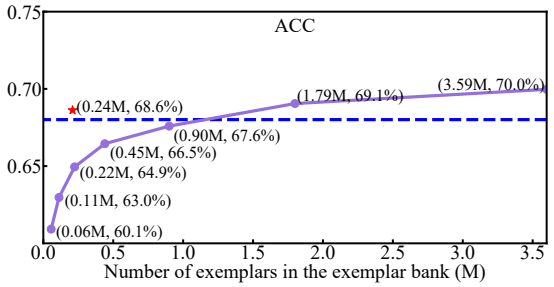

Figure 6: Scalability comparison. The curve indicate the accuracy according to exemplar number from synthesis.

Figure 7: Visualization of text recognition results.

Since the exemplars from the training set are limited, we use a standard text renderer (oh-my ocr 2021) to synthesize the Chinese text-line image. We also incorporate the efficient similarity search library faiss (Johnson, Douze, and Jégou 2019) to speed up the retrieval module. It takes 15.6ms for a single exemplar to retrieve in the exemplar bank containing more than 3 million exemplars, and meanwhile keeps a good recognition result. Figure 6 plots the performance versus the number of exemplars in the exemplar bank. With a similar bank size, the exemplars from real-world web images can reach higher accuracy than those from synthesized images (68% vs. 65%). Adding more synthesized exemplars boosts the accuracy and we obtain an accuracy of 70.0% with 3.59M exemplars in the bank.

Comparison with State-of-the-Arts

To evaluate the performance of our exemplar-based framework, we also compare ours with the state-of-the-art Chinese text recognition methods. Table 6 summarizes the recognition accuracies and normalized edit distances in the four types of Chinese characters. Among them, the first group contains several well-known recognition methods, such as CRNN (Shi, Bai, and Yao 2017), ASTER (Shi et al. 2018), and MORAN (Luo, Jin, and Sun 2019). These baseline performance numbers are from (Yu et al. 2021). From the table, we see that the proposed DECTR outperforms all the methods. We also compare ours with the state-of-the-art approaches shown in the middle part of Table 6, including the image processing based method TransOCR (Chen, Li, and Xue 2021), the effective visual model based scheme SVTR-L (Du et al. 2022), ABINet (Fang et al. 2021) that uses linguistic knowledge, and MaskOCR-ViT-B† (Lyu et al. 2022) that employ model pretraining. It clearly shows that our proposed DECTR-B outperforms TransOCR, SVTR-L, and ABINet methods on these datasets. The Chinese text recognition results of some examples are illustrated in Figure 7. The performance of our DECTR-L reaches the same magnitude as that of MaskOCR-ViT-B with vision-language pretraining, while the parameter size of our framework is significantly smaller than theirs.

Conclusions

In this paper, we introduced a novel deep exemplar-based framework for Chinese text recognition, which consists of the exemplar discovery network to select candidate character regions as exemplars, as well as the exemplar retrieval module that finds the most similar exemplars and propagates the character label. Compared with directly employing the results from classification head of the network, the proposed DECTR offers the ability to correct misrecognized characters and improves overall text recognition performance. Through an extensive set of Chinese text recognition experiments, we have shown that DECTR is more effective than existing recognition approaches, generating very competitive results for different types of real-world Chinese texts. As future work, we will explore topics such as employing efficient backbone (Meng et al. 2022) and applying DECTR in related applications (Du et al. 2023; Wu et al. 2024).

Acknowledgments

This work was supported by National Archives Administration of China Research Program (2024-X-013) and Shanghai Archives Research Program (Grant No. 2425). The computations in this research were performed using the CFFF platform of Fudan University.

References

Baek, J.; Kim, G.; Lee, J.; Park, S.; Han, D.; Yun, S.; Oh, S. J.; and Lee, H. 2019. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4715–4723.

Cao, Z.; Lu, J.; Cui, S.; and Zhang, C. 2020. Zero-shot handwritten Chinese character recognition with hierarchical decomposition embedding. *Pattern Recognition*, 107: 107488.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 213–229.

Chechik, G.; Sharma, V.; Shalit, U.; and Bengio, S. 2010. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(3).

Chen, J.; Li, B.; and Xue, X. 2021. Scene Text Telescope: Text-Focused Scene Image Super-Resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12026–12035.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1290–1299.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *IEEE/CVF International Conference on Computer Vision*, 764–773.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations*.

Du, X.; Ma, T.; Zheng, Y.; Ye, H.; Wu, X.; and He, L. 2020. Scene text recognition with temporal convolutional encoder. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2383–2387.

Du, X.; Zhou, Z.; Zheng, Y.; Wu, X.; Ma, T.; and Jin, C. 2023. Progressive scene text erasing with self-supervision. *Computer Vision and Image Understanding*, 233: 103712.

Du, Y.; Chen, Z.; Jia, C.; Yin, X.; Zheng, T.; Li, C.; Du, Y.; and Jiang, Y.-G. 2022. SVTR: Scene Text Recognition with a Single Visual Model. In *International Joint Conference on Artificial Intelligence*, 884–890.

Fang, S.; Xie, H.; Wang, Y.; Mao, Z.; and Zhang, Y. 2021. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7098–7107.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778.

Johnson, J.; Douze, M.; and Jégou, H. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3): 535–547.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Liu, C.; Yang, C.; and Yin, X.-C. 2022. Open-Set Text Recognition via Character-Context Decoupling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4523–4532.

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

Lyu, P.; Zhang, C.; Liu, S.; Qiao, M.; Xu, Y.; Wu, L.; Yao, K.; Han, J.; Ding, E.; and Wang, J. 2022. Maskocr: text recognition with masked encoder-decoder pretraining. *arXiv preprint arXiv:2206.00311*.

Meng, L.; Li, H.; Chen, B.-C.; Lan, S.; Wu, Z.; Jiang, Y.-G.; and Lim, S.-N. 2022. AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12309–12318.

Noh, H.; Araujo, A.; Sim, J.; Weyand, T.; and Han, B. 2017. Large-scale image retrieval with attentive deep local features. In *IEEE/CVF International Conference on Computer Vision*, 3456–3465.

oh-my ocr. 2021. text renderer. https://github.com/oh-my-ocr/text_renderer.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Peng, D.; Jin, L.; Ma, W.; Xie, C.; Zhang, H.; Zhu, S.; and Li, J. 2022. Recognition of handwritten Chinese text by segmentation: a segment-annotation-free approach. *IEEE Transactions on Multimedia*, 25: 2368–2381.

Radenović, F.; Tolias, G.; and Chum, O. 2018. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7): 1655–1668.

Sheshadri, K.; and Divvala, S. K. 2012. Exemplar Driven Character Recognition in the Wild. In *British Machine Vision Conference*, 1–10.

Shi, B.; Bai, X.; and Yao, C. 2017. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11): 2298–2304.

Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2018. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9): 2035–2048.

Souibgui, M. A.; Fornés, A.; Kessentini, Y.; and Megyesi, B. 2022. Few shots are all you need: A progressive learning approach for low resource handwritten text recognition. *Pattern Recognition Letters*, 160: 43–49.

Tan, F.; Yuan, J.; and Ordonez, V. 2021. Instance-level image retrieval using reranking transformers. In *IEEE/CVF International Conference on Computer Vision*, 12105–12115.

Weinzaepfel, P.; Lucas, T.; Larlus, D.; and Kalantidis, Y. 2022. Learning super-features for image retrieval. In *International Conference on Learning Representations*.

Wen, L.; Wang, Y.; Zhang, D.; and Chen, G. 2023. Visual Matching is Enough for Scene Text Retrieval. In *ACM International Conference on Web Search and Data Mining*, 447–455.

Wu, X.; Xiao, L.; Du, X.; Zheng, Y.; Li, X.; Ma, T.; Jin, C.; and He, L. 2024. Cross-domain document layout analysis using document style guide. *Expert Systems with Applications*, 245: 123039.

Yao, C.; Bai, X.; Shi, B.; and Liu, W. 2014. Strokelets: A learned multi-scale representation for scene text recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4042–4049.

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

Yu, H.; Wang, X.; Li, B.; and Xue, X. 2023. Chinese text recognition with a pre-trained clip-like model through image-ids aligning. In *IEEE/CVF International Conference on Computer Vision*, 11943–11952.

Yu, M.-M.; Zhang, H.; Yin, F.; and Liu, C.-L. 2024. An approach for handwritten Chinese text recognition unifying character segmentation and recognition. *Pattern Recognition*, 151: 110373.

Zhan, F.; and Lu, S. 2019. ESIR: end-to-end scene text recognition via iterative rectification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2059–2068.

Zhang, C.; Gupta, A.; and Zisserman, A. 2020. Adaptive text recognition through visual matching. In *European Conference on Computer Vision*, 51–67.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations*.