Knowledge-Enhanced Historical Document Segmentation and Recognition

En-Hao Gao[1,2], **Yu-Xuan Huang**[1,2], **Wen-Chao Hu**[1,2], **Xin-Hao Zhu**[1,2], **Wang-Zhou Dai**[1,3]

[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]School of Artificial Intelligence, Nanjing University, China
[3]School of Intelligence Science and Technology, Nanjing University, China
{gaoeh, huangyx, huwc, zhuxh, daiwz}@lamda.nju.edu.cn

Abstract

Optical Character Recognition (OCR) of historical document images remains a challenging task because of the distorted input images, extensive number of uncommon characters, and the scarcity of labeled data, which impedes modern deep learning-based OCR techniques from achieving good recognition accuracy. Meanwhile, there exists a substantial amount of expert knowledge that can be utilized in this task. However, such knowledge is usually complicated and could only be accurately expressed with formal languages such as first-order logic (FOL), which is difficult to be directly integrated into deep learning models. This paper proposes KESAR, a novel **K**nowledge-**E**nhanced Document **S**egmentation **A**nd **R**ecognition method for historical document images based on the Abductive Learning (ABL) framework. The segmentation and recognition models are enhanced by incorporating background knowledge for character extraction and prediction, followed by an efficient joint optimization of both models. We validate the effectiveness of KESAR on historical document datasets. The experimental results demonstrate that our method can simultaneously utilize knowledge-driven reasoning and data-driven learning, which outperforms the current state-of-the-art methods.

1 Introduction

Document image analysis is a common task that aims to extract text from document images. Typically, it involves two steps, image *segmentation* and *recognition*. Segmentation is devised to identify and isolate regions containing the desired texts. After segmentation, recognition transforms the segmented images into textual form. With the rapid development of OCR technologies, the analysis of modern document images can now be well addressed (Long, He, and Yao 2021), as these images often have neat arrangement, clear handwriting, and abundant labeled data. However, different from modern ones, the analysis of historical documents, including handwritten manuscripts and early prints, remains a challenging, unresolved issue.

Three main challenges hinder the segmentation and recognition of historical document images. Firstly, text lines are often distorted and densely packed, leading to substantial challenges for image segmentation. Secondly, historical documents often include a wide range of character categories. For instance, while modern Chinese documents typically use around 3,500 characters, historical counterparts may contain over 10,000 characters. This extensive character dictionary requires a large amount of labeled data for the recognition model training. Thirdly, annotating historical documents is time-consuming and requires a high level of expert knowledge. This results in a scarcity of labeled images, consequently leading to inferior performance of modern data-driven segmentation and recognition models.

However, humans are able to make successful segmentation and recognition from historical manuscripts by utilizing background knowledge, which is also a promising way for enhancing machine learning performance (Raedt et al. 2020). For instance, during the segmentation of Chinese document images, characters typically have a square-like shape. Furthermore, characters within the same text line are expected to exhibit similar aspect ratios and sizes. First-order logic (FOL) rules provide a precise way to express such knowledge. However, it is non-trivial to inject these rules into the learning process of common deep learning models, since the application of FOL typically relies on logical reasoning, a discrete process that is difficult to integrate with gradient-based numerical optimization methods.

In order to leverage human knowledge to empower document image analysis, we adopt the ABductive Learning (ABL) framework (Zhou 2019; Zhou and Huang 2022). This novel paradigm bridges data-driven machine learning and knowledge-driven logical reasoning while preserving the expressive power of both. In ABL, the machine learning model initially converts raw data into primitive logic facts, named pseudo-labels. The reasoning component then revises pseudo-labels that are inconsistent with the FOL knowledge base by abductive reasoning (Magnani 2009), a.k.a. abduction. Subsequently, these knowledge-refined pseudo-labels are utilized to update the machine learning model, and the above routine repeats iteratively.

In this paper, we propose KESAR (Knowledge-Enhanced document Segmentation And Recognition) to tackle the above challenges based on the ABL framework. It first trains the segmentation model with structural knowledge, where the predicted character regions and affinities (area between adjacent characters) are refined by the knowledge base via abduction. Then, to address the issue of label scarcity, it

leverages the proposed abductive matching mechanism to train the recognition model that is used to predict text for single-character images. In this process, a dynamic programming algorithm is utilized to conduct abductive matching efficiently. Finally, it employs joint optimization, allowing the segmentation and recognition models to mutually enhance their performance instead of being trained separately. In this process, we propose the *Over-Segmentation and Recombination (OSR)* algorithm, which enables the segmentation model to improve its performance by leveraging the recognition model's ability to differentiate characters.

To show the effectiveness of KESAR, we conduct extensive experiments on three datasets. These datasets include a substantial number of challenging images, featuring severe distortions, varying scales, and multiple sources of noise. Ablation studies demonstrate the importance of each component of our method, and empirical evaluations show that KESAR outperforms state-of-the-art OCR methods in both segmentation and recognition tasks.

2 Related Work

Recently, deep learning-based scene text detection methods have achieved remarkable results. They can be broadly classified into two categories: segmentation-based and regression-based methods. Typically, segmentation-based methods involve the integration of pixel-level predictions, followed by post-processing algorithms to derive the bounding boxes. CRAFT (Baek et al. 2019) predicts the probabilities of character regions and affinities for each pixel. PSENet (Wang et al. 2019b) proposes a progressive scale expansion mechanism, learning and enlarging text kernels to cover all text instances. Based on PSENet, PAN (Wang et al. 2019c) implements a pixel aggregation process by predicting the pixel similarities. Besides, regression-based methods try to predict the contours of text lines directly. FCENet (Zhu et al. 2021) regresses text lines on the Fourier domain and reconstructs contours during the inference stage. ABCNet (Liu et al. 2020) utilizes Bezier curves to parameterize polygon annotations, equipping the model with the ability to detect text lines of arbitrary shapes. However, the former types of models typically resort to weakly supervised training, potentially leading to inaccurate results given limited data. Meanwhile, the new contour representations heavily depend on highly specialized network architectures. In contrast, by introducing human knowledge, our method can sufficiently exploit the supervised information from limited labeled data, thus improving data efficiency. Besides, it imposes little limitation on the model's specific form.

Text recognition is another important component of document image analysis, which aims at recognizing text through a cropped text image. Some recognition approaches attempt to rectify irregular images to regular ones before recognition with an exemplary work of STN (Jaderberg et al. 2015). In contrast, DAN (Wang et al. 2020) and Robust Scanner (Yue et al. 2020) represent encoder-decoder-based methods, using the attention mechanism to capture neighborhood information, yielding promising results in irregular text recognition. Other approaches such as CA-FCN (Liao et al. 2019) and CCN (Xing et al. 2019) address recognition by segmenting each character to circumvent issues with irregular layouts. However, successful text recognition in these previous works usually requires substantial labeled data. This might be feasible for modern documents but presents significant challenges when dealing with historical documents.

The incorporation of human knowledge has long been considered an effective approach to addressing data scarcity. In recent years, advancements have been made in leveraging symbolic reasoning to enhance the performance of machine learning models such as neural networks, especially when certain domain knowledge is available. For instance, some approaches express logical domain knowledge as constraints within the neural network's loss function to guide the training process (Xu et al. 2018; Yang, Lee, and Park 2022). Other approaches endeavor to learn domain knowledge within neural networks using specialized layers (Wang et al. 2019a). Additionally, some methods interpret neural network outputs as probability distributions over symbols, subsequently invoking a symbolic system to derive solutions (Manhaeve et al. 2018; Tsamoura, Hospedales, and Michael 2021). Many of these methods use continuous functions to approximate logical constraints and discrete operators, which results in bias in the approximated inference and requires large amounts of training data. Our method is based on Abductive Learning (ABL) (Zhou 2019; Dai et al. 2019; Zhou and Huang 2022), a framework that bridges machine learning and symbolic reasoning via logical abduction. ABL has also demonstrated the capability to build a knowledge base from data (Huang et al. 2023a) or knowledge graph (Huang et al. 2023b). Following ABL, our method is capable of fully leveraging the deep learning capability for feature extraction from raw images, while also preserving the complete expressive power of logical reasoning for knowledge processing in symbolic space, which significantly improves the model performance.

3 Preliminaries

Abductive Reasoning. Abductive reasoning, a.k.a abduction, is a basic form of logical inference, which seeks an explanation for an observation. Formally, given observations O, based on background knowledge base KB, it generates a set of abducibles Δ consistent with KB and satisfies $KB \cup \Delta \models O$, where $\models$ stands for logical entailment. For example, when observing a text line, based on knowledge of text structure, we could explain that there are several characters with similar shapes and sizes in this text line.

Abductive Learning. The target of ABductive Learning (ABL) (Zhou 2019; Zhou and Huang 2022) is to train a machine learning model given unlabeled data and knowledge base. In ABL, the machine learning model perceives primitive logic facts from raw data, while logical abduction exploits the knowledge base to revise wrongly perceived facts to improve the machine learning model. For example, if a model predicts several bounding boxes for characters with dissimilar shapes within a text line, which are inconsistent with the knowledge base, ABL utilizes abduction to revise wrong bounding boxes and treats them as ground-truth labels to update the model.

4 The KESAR Approach

In this section, we first introduce an overview of the proposed model training method, KESAR (Knowledge-Enhanced historical document Segmentation And Recognition), and then present the details of its three learning stages.

4.1 Overview

KESAR consists of two machine learning models for image segmentation and recognition, respectively:

- **Segmentation Model**. The segmentation model takes the raw image as input, predicting the probability distribution for each pixel on three distinct categories: *character region* (an area containing a character), *character affinity* (an area between two adjacent characters), or *background*. The watershed algorithm is then employed to aggregate pixels that are likely to be situated within character regions, thereby isolating each single character. The images of the segmented characters then serve as inputs for the recognition model. On the other hand, character affinity is used to generate text lines by connecting dispersed character regions during the inference stage.
- **Recognition Model**. After the segmentation model identifies individual characters, the recognition model merely needs to predict text of single-character images. This task is relatively straightforward, and a small-scale ResNet network (He et al. 2016) can accomplish it effectively.

To integrate knowledge for document image segmentation and recognition, KESAR employs a three-stage learning methodology:

1. **Segmentation with Structural Knowledge**. It incorporates text structure knowledge to augment the weakly-supervised learning process of the segmentation model.
2. **Recognition with Abductive Matching**. It uses both knowledge of abductive matching and unlabeled cropped character images to train the recognition model.
3. **Joint Optimization**. It uses glyph knowledge learned by the recognition model to further improve the segmentation model's performance. Meanwhile, the refined character segmentation results also facilitate the learning of the recognition model in turn.

4.2 Segmentation with Structural Knowledge

Due to the difficulty of labeling, most historical document images only have line-level labels (bounding box and text of each line), whereas the learning of the segmentation model requires character-level supervised information.

To bridge this discrepancy, we adopt the abductive learning framework, which incorporates structural knowledge of characters to deal with limited supervision. Fig. 1 illustrates the learning pipeline of the segmentation model in the task of Chinese historical document segmentation. Firstly, the segmentation model takes the document image as input and predicts the character region to generate pseudo-character bounding boxes. Then, the reasoning component rectifies bounding boxes inconsistent with the knowledge base through abductive reasoning, given the text-line annotations. These revised character bounding boxes are then

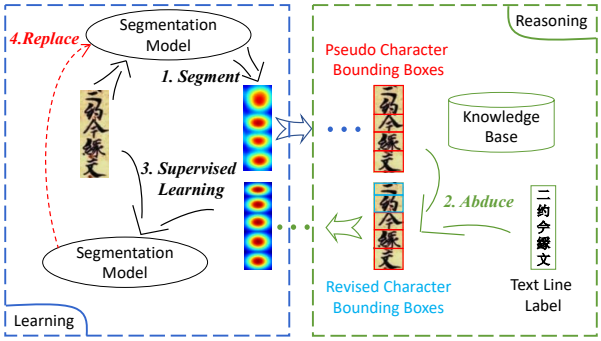


Figure 1: Illustration of Segmentation with Structural Knowledge. Each iteration begins with the character region prediction to extract pseudo-bounding boxes. Then, it employs abduction to revise inconsistent bounding boxes based on the knowledge base. Finally, it generates revised character regions to update the segmentation model, which will replace the origin one after each iteration.

used to update the segmentation model. In this example, we can precisely formalize human knowledge of text structure to FOL rules as follows:

$$\text{reg\_bbox}(B) \leftarrow \text{close}(\text{asp\_rat}(B), 1). \tag{1}$$

$$\begin{aligned}\text{reg\_textline}(TL) \leftarrow\ & \text{bbox\_seq}(TL) = \{B_1, B_2, \ldots\} \\ & \wedge\ \text{reg\_bbox}(B_1) \wedge \text{reg\_bbox}(B_2) \wedge \ldots \\ & \wedge\ \text{close}(\text{asp\_rat}(B_1), \text{asp\_rat}(B_2), \ldots) \\ & \wedge\ \text{close}(\text{size}(B_1), \text{size}(B_2), \ldots).\end{aligned} \tag{2}$$

$$\begin{aligned}\text{false} \leftarrow\ & \text{bbox\_seq}(TL) = \{B_1, B_2, \ldots\} \\ & \wedge \text{horizontal\_adjacent}(B_1, B_2, \ldots).\end{aligned} \tag{3}$$

"$\leftarrow$" is implication, which means that if premises on the right hold, then the conclusion on the left holds; $\text{reg\_bbox}(B)$ is the regular-shape constraint on bounding box B; $\text{reg\_textline}(TL)$ determines whether a sequence of character bounding boxes, $\text{bbox\_seq}(TL) = \{B_1, B_2, \ldots\}$, contained in TL composes a regular text line; $\text{close}(V_1, V_2, ...)$ calculates the variance of its arguments to assess whether the arguments are adequately proximate; $\text{asp\_rat}(B)$ and $\text{size}(B)$ calculate aspect ratio and size of a bounding box B, respectively. These FOL rules essentially convey three fundamental aspects of background knowledge in Chinese historical documents segmentation:

(1) Chinese characters are square-shaped.

(2) Characters within the same text line possess similar aspect ratios and sizes.

(3) Vertically-aligned text does not contain horizontally adjacent characters.

In this task, abductive reasoning aims to revise character bounding boxes by maximizing a consistency measure that quantifies the degree to which these boxes align with the semantics of predicates (e.g., close, reg_bbox) in rules

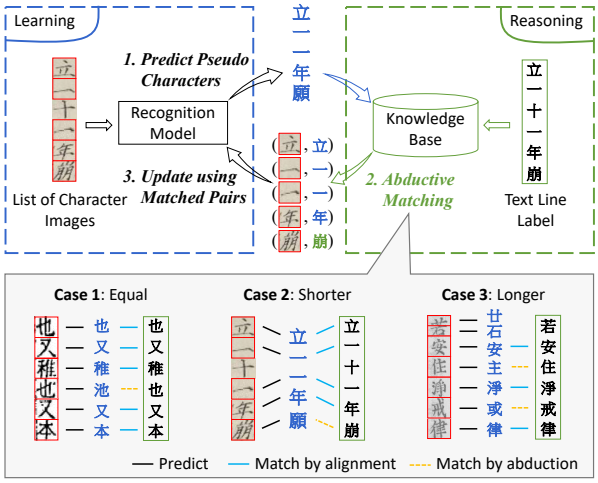


Figure 2: The upper half illustrates the learning pipeline of the recognition model. In each iteration, the recognition model first predicts pseudo-character labels from images. Potential inaccuracies are then rectified by the knowledge base via abductive matching. The model is then updated using these matched character images and labels. The lower half demonstrates three typical types of matching relationships between prediction and target in abductive matching. Yellow dotted lines represent the results of abduction, which are propagated from the blue ones.

(1)-(3). The consistency measure can be calculated in various way (Huang et al. 2020, 2021), and in KESAR it is the negative weighted sum of the predicates' value and the distance between the abduced and predicted boxes. Among all consistent bounding box sets, the one with the closest aspect ratio and size would have the maximal consistency.

Considering the measure's non-convexity, revising character bounding boxes by directly maximizing the consistency is time-consuming. In practice, the process begins with merging horizontally adjacent boxes. Then, irregular boxes are identified by examining the width and height of the text line. Finally, these irregular boxes are revised according to the average size of regular ones.

4.3 Recognition with Abductive Matching

The recognition model takes character images as input and predicts the text. Although most historical document images only have text-line labels, the predicted character bounding boxes of the segmentation model can be used to generate training images. Nonetheless, a significant discrepancy remains when utilizing these images to train the recognition model, primarily due to the potential omission or misidentification of character bounding boxes. As shown in the upper half of Fig. 2, the input text-line image contains 6 characters, while the segmentation model only predicts 5 bounding boxes. Since we do not know the correspondence between the text-line label and these bounding boxes, annotating such 5 boxes with the 6 ground-truth characters becomes puzzling, especially when the recognition result is incorrect.

Algorithm 1: Abductive Matching

Input: Predicted string $P = (p_1, p_2, \ldots, p_n)$; Ground-truth string $G = (g_1, g_2, \ldots, g_m)$; Maximum length of matched substrings max_len

Output: Maximum set of groundings Δ

Initialization: $p_0 \leftarrow$ '[START]'; $g_0 \leftarrow$ '[START]'; $res \leftarrow [[0], \ldots, [0]]$; $trace \leftarrow [[0], \ldots, [0]]$
for $i = 1$ **to** n **do**
  for $j = 1$ **to** m **do**
    if $p_i = g_j$ **then**
      $res[i, j] \leftarrow res[i-1, j-1] + 1$
      $trace[i, j] \leftarrow 1$
      for $k = 1$ **to** $min(i, j, max\_len)$ **do**
        if $p_{i-k} = g_{j-k}$ **then**
          $res[i, j] \leftarrow res[i-k, j-k] + k$
          $trace[i, j] \leftarrow k$
          $break$
    else
      $res[i, j] \leftarrow max(res[i-1, j], res[i, j-1])$
$\Delta \leftarrow postprocess(res, trace)$

To address this challenge, we introduce the second stage learning of KESAR, namely, Recognition with Abductive Matching. By incorporating general human knowledge for matching relationship construction, this approach can generate pairs of matched character images and labels, thereby promoting the learning of the recognition model.

The learning process also follows the framework of ABL. This time, the medium facilitating the interaction between learning and reasoning changes from character bounding boxes to character labels. The upper half of Fig. 2 shows the closed-loop learning process of the recognition model. During each cycle, the predicted characters are revised by the knowledge base via abductive matching and these refined characters, paired with input images, are then used to train the recognition model.

The core strategy of abductive matching involves initially aligning identical segments between the prediction and the text-line label and subsequently propagating the matching relationships to equal-length intervals. The lower half of Fig. 2 illustrates three cases of abductive matching, each representing a length relationship between the prediction and the text-line label. The first case often occurs when segmentation is accurate, but recognition may be erroneous. The latter two cases commonly result from the omission and incorrect identification of character bounding boxes, respectively.

Efficient Optimization. In abductive matching, different alignment ways will lead to different propagation results and hence different numbers of matched character images and labels. To generate more training data for the recognition model, we construct the following optimization problem:

$$\begin{aligned} \max \quad & \left|\{(p_j^i, g_j^i) | \ \text{match\_char}(p_j^i, g_j^i)\}\right| \\ \text{s.t.} \quad & \text{match\_str}(P_i, G_i) \\ \text{where} \quad & P_i = \{p_1^i, p_2^i, \ldots, p_l^i\}, G_i = \{g_1^i, g_2^i, \ldots, g_l^i\}, \\ & \text{substr}(P_i, P), \text{substr}(G_i, G), l < max\_len. \end{aligned}$$

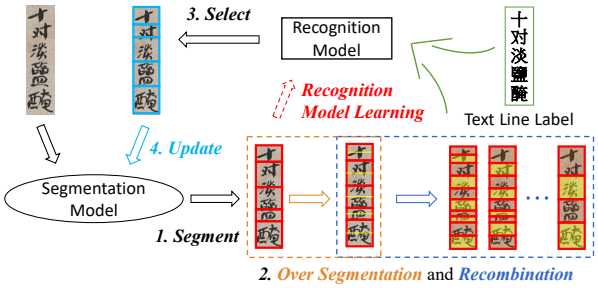


Figure 3: Illustration of Joint Optimization of the segmentation and recognition models. Images segmented by the segmentation model will be used in the recognition model learning, as described in Section 4.3. Furthermore, these images will be used to train the segmentation model after refinement by the Over-Segmentation and Recombination algorithm. Over-segmentation divides the segmented images into finer segments, and recombination fuses these split images in various ways. After being assessed by the recognition model, the recombination way with the highest score will be subsequently used to update the segmentation model.

where $|\cdot|$ calculates the number of elements in the set; $\text{match\_str}(P_i, G_i)$ holds true when P_i and G_i have equal length and their respective first and last characters are also identical; $\text{match\_char}(p_j^i, g_j^i)$ holds true when p_j^i and g_j^i are in the same position of a pair of matched strings; P, G are the predicted string and the target string respectively; $\text{substr}(P_i, P)$ holds true when P_i is a substring of P; max_len restricts the length of matched substrings, since the credibility of the matching established by propagation gradually decreases as the length of the substring increases.

Although this is a combinatorial optimization problem, it can be solved by a dynamic programming algorithm with polynomial time complexity $O(nm * max\_len)$, where n and m are lengths of P and G respectively. Algorithm 1 represents the detail of the proposed ABductive Matching (ABM) method. The algorithm calculates each element in the res array in ascending order. For $res[i, j]$, if p_i and g_j are the same, they can be matched together and $res[i, j]$ is initialized as $res[i-1, j-1] + 1$. Then, the algorithm traverses forward up to max_len steps to find another pair of matching characters, so that the characters in between can be matched through abduction and $res[i, j]$ can be updated correspondingly (cf. Line 7-11 in Algorithm 1). If p_i and g_j are different, then $res[i, j]$ is set to the maximum value of $res[i-1, j]$ and $res[i, j-1]$ (cf. Line 13 in Algorithm 1).

4.4 Joint Optimization

The joint optimization focuses on using glyph knowledge learned by the recognition model to further improve the predictive accuracy of the segmentation model, especially in complex scenarios where characters are closely packed or portions of a single character are distinctly separated. Besides the performance improvement of the segmentation model, this augmented character segmentation capability also boosts the learning of the recognition model.

Algorithm 2: Over-Segmentation and Recombination (OSR)

Input: Recognition model f; Sequence of bounding boxes $B = (B_1, B_2, \ldots, B_n)$; Sequence of character labels $C = (c_1, c_2, \ldots, c_m)$; Max recombination number r
Output: Sequence of recombined bounding boxes D

1: Initialization:$D \leftarrow \emptyset$; $score \leftarrow [0]$; $match\_len \leftarrow [0]$; $trace \leftarrow [0]$
2: $OB \leftarrow OverSegment(B)$
3: **for** $i = 1$ **to** $2n$ **do**
4: $\quad score.append(0)$
5: $\quad$ **for** $j = max(1, i - r + 1)$ **to** $2n$ **do**
6: $\quad\quad tar\_char = C[match\_len[j-1] + 1]$
7: $\quad\quad comb\_score = f(Comb(OB[j:i]), tar\_char)$
8: $\quad\quad new\_score = score[j-1] + comb\_score$
9: $\quad\quad$ **if** $new\_score > score[i]$ **then**
10: $\quad\quad\quad score[i] \leftarrow new\_score$
11: $\quad\quad\quad match\_len[i] \leftarrow match\_len[j-1] + 1$
12: $\quad\quad\quad trace[i] \leftarrow j - 1$
13: $end\_index \leftarrow \arg\max_{i, match\_len[i]=m} score[i]$
14: $D \leftarrow postprocess(trace, end\_index)$

Figure 3 presents the overall pipeline of joint optimization. The red dotted arrow represents the recognition model learning process established in Section 4.3. To make the segmentation model learning benefit from the recognition model, we need to close the loop. Considering that the primary issue with the segmentation model is the incorrect merging or splitting of characters, we propose the Over-Segmentation and Recombination (OSR) algorithm.

Algorithm 2 presents details of the OSR approach. Initially, the algorithm segments each bounding box in B into two new vertically packed bounding boxes, where the segmentation point is approximately halfway through the height of the original character bounding box (cf. Line 2 in Algorithm 2). Following this, OSR iteratively processes these segmented boxes, merging those at the sequence's tail and employing the recognition model to assess the effect of the combination (cf. Line 3-12 in Algorithm 2). The objective is to find the recombined bounding box sequence with the highest score. This can be computed recursively from $trace$ and end_index. The recombination process is inspired by the idea of Rg-ABBS (Xie et al. 2019). However, Rg-ABBS utilizes beam search to determine the combination path, any rejection of the (partial) optimal solution will remove the global optima from subsequent searches. In contrast, our method restricts the number of combined boxes, as a single character will not be excessively lengthy in the majority of cases, thereby ensuring both efficiency and accuracy.

5 Experiments

This section presents the experimental results on three historical document datasets to demonstrate the effectiveness of each stage within KESAR and compare it with state-of-the-art methods. All experiments are conducted on a server with 8 Nvidia V100 GPUs. The code is available for download[1].

[1]https://github.com/AbductiveLearning/ABL-HD

Method	MTH			GBACHD			TKH		
	R(%)	P(%)	F(%)	R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
PSENet	88.4	87.3	87.8	73.2	82.2	77.5	**97.5**	90.7	94.0
FCENet	78.8	81.6	80.2	71.1	69.0	70.0	84.1	84.8	84.4
CRAFT	85.9	93.4	89.5	82.5	93.2	87.5	97.3	**98.3**	**97.8**
KESAR (w/o JOPT)	92.1	**94.0**	93.1	93.5	93.4	93.5	-	-	-
KESAR	**93.1**	93.4	**93.2**	**94.4**	**94.0**	**94.2**	-	-	-

Table 1: Segmentation model comparison results on MTH, GBACHD, and TKH. R, P, and F represent the recall, precision, and F-measure respectively. The performances of KESAR (w/o JOPT) and KESAR on TKH are neglected since character-level annotations of TKH are used in the pre-training and there is no need to revise these annotations by structural knowledge or joint optimization. Comparison on TKH is for reference only since the training label used by CRAFT is different from others.

5.1 Datasets

TKH (Yang et al. 2018) is a collection of historical documents released by HCIILAB, containing 1,000 images sourced from the Tripitaka Koreana. The dataset incorporates both character and text-line annotations. Text lines are neatly arranged, characters are relatively uniform in size, and the variety of character types is somewhat limited. Due to the inclusion of character annotations, we randomly select 600 images to serve as the pre-training data for KESAR.

MTH (Ma et al. 2020) is a more challenging historical document dataset, characterized by prevalent text line distortions, intricate page layouts, and occasional inclusion of drawings. Comprising 2,200 images, the dataset is randomly partitioned into training and testing subsets at a 7:3 ratio. Although the MTH dataset encompasses both character and text-line annotations, only the latter are employed to validate the efficiency of our method.

GBACHD is the most challenging dataset in our experiments, released in the 2022 Greater Bay Area (Huangpu) International Algorithm Case Competition. Encompassing 2,000 images, the dataset features over 15,000 character categories, embracing numerous rare characters and variant forms. GBACHD provides only text-line annotations. The complexities of the segmentation and recognition tasks derive not only from the severe distortions and varying scales but also from the presence of multiple sources of noise such as stains, blurred notes, and seals. The dataset is randomly partitioned into 1,400 images designated for training, with the remaining images set aside for testing.

5.2 Implementation Details

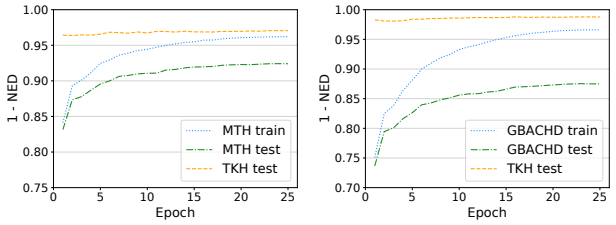
Our baseline segmentation model is CRAFT (Baek et al. 2019) with ResNet50 as its backbone. We first employ the training data of TKH to pre-train the segmentation model for 320 epochs and then utilize MTH and GBACHD to fine-tune the model for 180 and 80 epochs, respectively. Our recognition model is ResNet34. We first employ the training data of TKH to pre-train the network for 25 epochs and then utilize character images generated by the segmentation model to fine-tune the model for another 25 epochs. The Joint Optimization stage requires only 10 epochs. The whole training process can be finished in 15 hours. More implementation details are listed in the appendix.

5.3 Ablation Study

Influence of Structural Knowledge. We investigate the effect of structural knowledge on the segmentation model by comparing its performance with and without the utilization of a knowledge base for rectifying pseudo-character bounding boxes. We denote the structural knowledge-enhanced CRAFT model as KESAR (w/o JOPT) since it has not been fine-tuned by the joint optimization process. As shown in Table 1, KESAR (w/o JOPT) surpasses the vanilla CRAFT in terms of text line segmentation recall, precision, and F-measure. Notably, the F-measure (93.1%) achieved by KESAR (w/o JOPT) surpasses that of CRAFT by an absolute 3.6% on the MTH dataset and by an absolute 6.0% on the GBACHD dataset. Furthermore, since KESAR primarily serves as a model training method, it maintains the same inference speed as its baseline model, CRAFT. The performance of KESAR (w/o JOPT) on the TKH dataset is not included since the character-level annotations used in the training obviate the need for revision through knowledge and joint optimization.

Influence of Abductive Matching. We study the effect of abductive matching on the recognition model's performance. Our evaluation metrics include 1-N.E.D. (normalized edit distance) (Zhang et al. 2019) and the successful abduction rate, defined as the proportion of correctly abduced labels. A higher value of 1-N.E.D. indicates better recognition performance and a higher successful abduction rate means a larger portion of character images are matched with a character label. Fig. 4 illustrates the performance trajectory throughout the training process, demonstrating rapid improvement in recognition accuracy that eventually reaches a relatively high level of performance.

As shown in Table 2, KESAR achieves 0.924 1-N.E.D. on the MTH dataset, which is absolute 0.093 higher than the initial performance. Improvement is even more significant on the GBACHD dataset, where KESAR ultimately achieves 0.875 1-N.E.D., compared to the initial performance of 0.737. As also shown in Table 3, the trend of the successful abduction rate mirrors that of 1-N.E.D. and even converges more rapidly. It finally achieves a near-optimal result, where almost all character labels are matched with a character image. Since the recognition model of KESAR is a small-scale ResNet, its inference is highly efficient.



(a) MTH (b) GBACHD

Figure 4: Learning curves of the recognition model. Blue curves indicate the 1-N.E.D. evaluation results on the training data and green curves indicate results on the test data. Considering that TKH is employed during the pre-training phase, we also display results from the TKH test data using yellow curves.

Influence of Joint Optimization. We investigate the impact of joint optimization and focus on the performance improvement of the segmentation model, as the capability of recognition is highly dependent on the segmentation. As shown in Table 1, the performances of KESAR and KESAR (w/o JOPT) on the MTH dataset are comparable. This is primarily because the challenges posed by the MTH dataset stem from complex page layouts and variations in character scale across the image, whereas characters within the same text line are generally clearly separated. On the GBACHD dataset, there is a noticeable improvement in model performance, with KESAR surpassing KESAR (w/o JOPT) across all three metrics. This enhancement aligns with our expectations, given that many characters in the GBACHD images are densely clustered, and the recognition model aids in segmenting these difficult instances.

5.4 Comparisons with State-of-the-Art Methods

Text Line Segmentation. We employ PSENet (Wang et al. 2019b) and FCENet (Zhu et al. 2021) as comparison methods, which are implemented by the mmocr codebase (Kuang et al. 2021). By incorporating a progressive scale expansion mechanism and multi-scale kernels, PSENet is able to gradually expand the predicted text-line region, which enables the model to effectively distinguish densely packed text lines. Benefiting from the expressive power of Fourier Transformation to represent closed contours, FCENet is especially good at detecting highly distorted text lines which are prevalent in the GBACHD dataset. Table 1 summarizes our results including text line segmentation recall, precision, and F-measure on the MTH and GBACHD datasets. On MTH, KESAR surpasses all comparison models. We can find that the F-measure (93.2%) achieved by KESAR is 5.4% higher than PSENet and 13.0% higher than FCENet on the F-measure. The GBACHD dataset is significantly more challenging than the MTH dataset, featuring a higher level of noise and distortions. As a result, we find a substantial decrease in the performance of comparison methods. Nevertheless, KESAR still performs well, achieving the highest F-measure of 94.2%.

Method	MTH	GBACHD	TKH
RobustScanner	0.905	0.722	0.991 / **0.989**
ABINet	0.900	0.718	**0.992** / 0.988
KESAR	**0.924**	**0.875**	0.970 / 0.988

Table 2: Recognition model comparison results on MTH, GBACHD, and TKH. The performance metric is 1-N.E.D..

Epoch	1	2	4	10	15	25
MTH	0.949	0.989	0.992	0.994	0.995	0.996
GBACHD	0.919	0.977	0.983	0.988	0.990	0.992

Table 3: Rate of successful abduction w.r.t. training epoch on MTH and GBACHD.

Text Recognition. We utilize RobustScanner (Yue et al. 2020) and ABINet (Fang et al. 2021) as comparison methods, which are also implemented by the mmocr codebase. Since our method utilizes TKH for pre-training in the experiments on MTH and GBACHD, we include TKH in the training data of other methods for a fair comparison. Therefore, results on TKH represent the performance of models trained on both MTH/GBACHD and TKH, while tested solely on TKH. Our evaluation metric is 1-N.E.D.. As shown in Table 2, KESAR achieves superior performance on MTH and GBACHD. Remarkably on GBACHD, KESAR outperforms other methods by at least 0.153 1-N.E.D.. On TKH, comparison methods exhibit excellent performance and our method is competitive. It is worth noting that our training data comprise predicted text lines generated by the segmentation model, whereas comparative methods use ground-truth text lines as training data.

6 Conclusion

To exploit human knowledge in document segmentation and recognition, we propose a novel approach based on the abductive learning framework, aiming at using background knowledge to enhance character extraction and prediction performance. In detail, our method enables the model to refine segmentation results by utilizing structural knowledge, and the proposed abductive matching mechanism can generate character-level training data for the recognition model from text-line labels. Moreover, through joint optimization, the segmentation and recognition models can mutually benefit and enhance each other's performance. Empirical evaluation validates that our learning approach can significantly improve the performance of both segmentation and recognition models, outperforming the state-of-the-art OCR methods. KESAR is a general-purposed approach with sufficient flexibility in implementation, e.g., the segmentation and recognition models can be replaced by other networks and the knowledge base can be modified to adapt to other application scenarios.

Acknowledgments

This research was supported by NSFC (62206124) and JiangsuSF (BK20232003).

References

Baek, Y.; Lee, B.; Han, D.; Yun, S.; and Lee, H. 2019. Character region awareness for text detection. In *CVPR*, 9365–9374.

Dai, W.-Z.; Xu, Q.; Yu, Y.; and Zhou, Z.-H. 2019. Bridging machine learning and logical reasoning by abductive learning. *NeurIPS*.

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 *CVPR*, 7098–7107.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Huang, Y.-X.; Dai, W.-Z.; Cai, L.-W.; Muggleton, S. H.; and Jiang, Y. 2021. Fast Abductive Learning by Similarity-based Consistency Optimization. In *NeurIPS*, 26574–26584.

Huang, Y.-X.; Dai, W.-Z.; Jiang, Y.; and Zhou, Z.-H. 2023a. Enabling Knowledge Refinement upon New Concepts in Abductive Learning. In *AAAI*, 7928–7935.

Huang, Y.-X.; Dai, W.-Z.; Yang, J.; Cai, L.-W.; Cheng, S.; Huang, R.; Li, Y.-F.; and Zhou, Z.-H. 2020. Semi-Supervised Abductive Learning and Its Application to Theft Judicial Sentencing. In *ICDM*, 1070–1075.

Huang, Y.-X.; Sun, Z.; Li, G.; Tian, X.; Dai, W.-Z.; Hu, W.; Jiang, Y.; and Zhou, Z.-H. 2023b. Enabling Abductive Learning to Exploit Knowledge Graph. In *IJCAI*, 3839–3847.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; and Kavukcuoglu, K. 2015. Spatial Transformer Networks. In *NeurIPS*, 2017–2025.

Kuang, Z.; Sun, H.; Li, Z.; Yue, X.; Lin, T. H.; Chen, J.; Wei, H.; Zhu, Y.; Gao, T.; Zhang, W.; Chen, K.; Zhang, W.; and Lin, D. 2021. MMOCR: A Comprehensive Toolbox for Text Detection, Recognition and Understanding. In *ACM MM*, 3791–3794.

Liao, M.; Zhang, J.; Wan, Z.; Xie, F.; Liang, J.; Lyu, P.; Yao, C.; and Bai, X. 2019. Scene text recognition from two-dimensional perspective. In *AAAI*, 8714–8721.

Liu, Y.; Chen, H.; Shen, C.; He, T.; Jin, L.; and Wang, L. 2020. ABCNet: Real-Time Scene Text Spotting With Adaptive Bezier-Curve Network. In *CVPR*, 9806–9815.

Long, S.; He, X.; and Yao, C. 2021. Scene Text Detection and Recognition: The Deep Learning Era. *International Journal of Computer Vision*, 129(1): 161–184.

Ma, W.; Zhang, H.; Jin, L.; Wu, S.; Wang, J.; and Wang, Y. 2020. Joint Layout Analysis, Character Detection and Recognition for Historical Document Digitization. *ICFHR*, 31–36.

Magnani, L. 2009. *Abductive Cognition: The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning*. Springer-Verlag.

Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2018. DeepProbLog: Neural Probabilistic Logic Programming. In *NeurIPS*, 3749–3759.

Raedt, L. D.; Dumancic, S.; Manhaeve, R.; and Marra, G. 2020. From Statistical Relational to Neuro-Symbolic Artificial Intelligence. In *IJCAI*, 4943–4950.

Tsamoura, E.; Hospedales, T.; and Michael, L. 2021. Neural-symbolic integration: A compositional perspective. In *AAAI*, 5051–5060.

Wang, P.-W.; Donti, P.; Wilder, B.; and Kolter, Z. 2019a. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *ICML*, 6545–6554.

Wang, T.; Zhu, Y.; Jin, L.; Luo, C.; Chen, X.; Wu, Y.; Wang, Q.; and Cai, M. 2020. Decoupled attention network for text recognition. In *AAAI*, 12216–12224.

Wang, W.; Xie, E.; Li, X.; Hou, W.; Lu, T.; Yu, G.; and Shao, S. 2019b. Shape robust text detection with progressive scale expansion network. In *CVPR*, 9336–9345.

Wang, W.; Xie, E.; Song, X.; Zang, Y.; Wang, W.; Lu, T.; Yu, G.; and Shen, C. 2019c. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *CVPR*, 8440–8449.

Xie, Z.; Huang, Y.; Jin, L.; Liu, Y.; Zhu, Y.; Gao, L.; and Zhang, X. 2019. Weakly supervised precise segmentation for historical document images. *Neurocomputing*, 350: 271–281.

Xing, L.; Tian, Z.; Huang, W.; and Scott, M. R. 2019. Convolutional character networks. In *CVPR*, 9126–9136.

Xu, J.; Zhang, Z.; Friedman, T.; Liang, Y.; and Broeck, G. 2018. A semantic loss function for deep learning with symbolic knowledge. In *ICML*, 5502–5511.

Yang, H.; Jin, L.; Huang, W.; Yang, Z.; Lai, S.; and Sun, J. 2018. Dense and Tight Detection of Chinese Characters in Historical Documents: Datasets and a Recognition Guided Detector. *IEEE Access*, 6: 30174–30183.

Yang, Z.; Lee, J.; and Park, C. 2022. Injecting logical constraints into neural networks via straight-through estimators. In *ICML*, 25096–25122.

Yue, X.; Kuang, Z.; Lin, C.; Sun, H.; and Zhang, W. 2020. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *ECCV*, 135–151.

Zhang, R.; Zhou, Y.; Jiang, Q.; Song, Q.; Li, N.; Zhou, K.; Wang, L.; Wang, D.; Liao, M.; Yang, M.; Bai, X.; Shi, B.; Karatzas, D.; Lu, S.; and Jawahar, C. V. 2019. ICDAR 2019 Robust Reading Challenge on Reading Chinese Text on Signboard. In *ICDAR*, 1577–1581.

Zhou, Z.-H. 2019. Abductive learning: towards bridging machine learning and logical reasoning. *Science China Information Sciences*, 62(7): 76101.

Zhou, Z.-H.; and Huang, Y.-X. 2022. Abductive Learning. In Hitzler, P.; and Sarker, M. K., eds., *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 353–369. Amsterdam: IOS Press.

Zhu, Y.; Chen, J.; Liang, L.; Kuang, Z.; Jin, L.; and Zhang, W. 2021. Fourier contour embedding for arbitrary-shaped text detection. In *CVPR*, 3123–3131.