Rethinking Text Segmentation: A Novel Dataset and A Text-Specific Refinement Approach

Xingqian Xu¹, Zhifei Zhang², Zhaowen Wang², Brian Price², Zhonghao Wang¹, Humphrey Shi³,¹

¹UIUC, ²Adobe Research, ³University of Oregon

Abstract

Text segmentation is a prerequisite in many real-world text-related tasks, e.g., text style transfer, and scene text removal. However, facing the lack of high-quality datasets and dedicated investigations, this critical prerequisite has been left as an assumption in many works, and has been largely overlooked by current research. To bridge this gap, we proposed TextSeg, a large-scale fine-annotated text dataset with six types of annotations: word- and character-wise bounding polygons, masks, and transcriptions. We also introduce Text Refinement Network (TexRNet), a novel text segmentation approach that adapts to the unique properties of text, e.g., non-convex boundary, diverse texture, etc., which often impose burdens on traditional segmentation models. In our TexRNet, we propose text-specific network designs to address such challenges, including key features pooling and attention-based similarity checking. We also introduce trimap and discriminator losses that show significant improvement in text segmentation. Extensive experiments are carried out on both our TextSeg dataset and other existing datasets. We demonstrate that TexRNet consistently improves text segmentation performance by nearly 2% compared to other state-of-the-art segmentation methods. Our dataset and code can be found at https://github.com/SHI-Labs/Rethinking-Text-Segmentation.

1. Introduction

Text segmentation is the foundation of many text-related computer vision tasks. It has been studied for decades as one of the major research directions in computer vision, and it continuously plays an important role in many applications [2, 55, 56, 47, 9]. Meanwhile, the rapid advances of deep neural nets in recent years promoted all sorts of new text-related research topics, as well as new vision challenges on text. Smart applications, such as font style transfer, scene text removal, and interactive text image editing, require effective text segmentation approaches to parse text accurately from complex scenes. Without any doubt, text segmentation is critical for industrial usages because it could upgrade the traditional text processing tools to be more intelligent and automatic, relaxing tedious efforts on manually specifying text regions.

However, modern text segmentation has been left behind in both datasets and methods. The latest public text segmentation challenge was in 2013-2015, hosted by ICDAR [26]. Since then, three datasets: Total-Text [10], COCO_Ts [5], and MLT_S [6], were introduced. However, Total-Text is limited in scale, and the labeling quality in COCO_Ts and MLT_S needs further improvement (Figure 5). Moreover, all the three datasets contain only common scene text, discouraging text in other visual conditions, e.g., artistic design and text effects. As a result, these datasets do not meet modern research standards, such as large-scale and fine-annotated. Thus, we propose a new text segmentation dataset: TexSeg, that collects images from a wider range of sources, including both scene and design text, and with a richer set of accurate annotations. This dataset would lead to further advancements in text segmentation research.

Additionally, text segmentation algorithms and methods
in recent years fall behind other research topics, partially due to the lack of a proper dataset. Unlike the rapid advances in other segmentation research, only a few studies [46, 6, 14] have brought new text segmentation ideas. Meanwhile, these studies did not provide an intuitive comparison with modern SOTA segmentation approaches and were unable to demonstrate their advantages over other techniques. As aforementioned, effective text segmentation models are valuable in applications. With our strong motivation to bridge this gap, we propose Text Refinement Network (TexRNet), and we thoroughly exam its performance on five text segmentation datasets including the proposed TextSeg dataset. The details of our design principles and our network structure are given in Session 3, and experiments and ablations studies are shown in Session 5.

In summary, the main contributions of this paper are in three-folds:

- We introduce a new large-scale fine-annotated text segmentation dataset, TextSeg, consisting of 4,024 text images, including scene text and design text with various artistic effects. TextSeg has six types of annotations for each image, i.e., word- and character-wise quadrilateral bounding polygons, pixel-level masks, and transcriptions. TextSeg surpasses prior datasets on these aspects: 1) more diverse text fonts/styles from diverse sources/collections, 2) more comprehensive annotations, and 3) more accurate segmentation masks.

- We provide a new text segmentation approach, Text Refinement Network (TexRNet), aiming to solve the unique challenges from text segmentation. We design effective network modules (i.e., key features pooling and attention-based similarity checking) and losses (i.e., trimap loss and glyph discriminator) to tackle those challenges, e.g., diverse texture and arbitrary scales/shapes.

- Exhaustive experiments are conducted to demonstrate the effectiveness of the proposed TexRNet, which outperforms SOTA on our TextSeg and on another four representative datasets. Besides, we give prospects for downstream applications that could significantly benefit from text segmentation.

2. Related Work

2.1. Segmentation in Modern Research

Semantic and instance segmentation are popular tasks for modern research. In semantic segmentation, pixels are categorized into a fixed set of labels. Datasets such as PASCAL VOC [15], Cityscapes [12], COCO [34], and ADE20K [61] are frequently used in this task. Traditional graph models, e.g., MRF [31] and CRF [29], predict segments by exploring inter-pixel relationship. After CNNs became popular [28], numerous deep models were proposed using dilated convolutions [60, 7, 8, 50], encoder-decoder structures [44, 60, 8, 33], and attention modules [51, 48, 16]. Instance segmentation methods predict distinct pixel labels for each object instance. These methods can be roughly categorized into top-down approaches [20, 32, 21, 35, 53, 27] and bottom-up approaches [4, 17, 37, 54, 40]. Top-down approaches are two-stage methods that first locate object bounding boxes and then segment object masks within those boxes. Bottom-up approaches locate keypoints [57, 40] and find edges and affinities [17, 37, 54, 4] to assist the segmentation process.

2.2. Text Segmentation

Early methods frequently used thresholding [41, 45] for segmentation particularly on document text images. Yet such methods cannot produce satisfactory results on scene text images with complex colors and textures. Other ap-
proaches used low-level features \cite{36, 52, 3} and Markov Random Field (MRF) \cite{38} to bipartite scene text images. In \cite{36}, text features created from edge density/orientation were fed into a multiscale edge-based extraction algorithm for segmentation. In \cite{52}, a two-stage method was introduced in which foreground color distribution from stage one was used to refine the result for stage two. In \cite{3}, seed points of both text and background were extracted from low-level features and were later used in segmentation. Inspired by MRF, \cite{38} formulated pixels as random variables in a graph model, and then graph-cut this model with two pre-selected seeds. In recent years, several deep learning methods \cite{46, 14, 6} were proposed for text segmentation. The method proposed by \cite{46} is a three-stage CNN-based model, in which candidate text regions were detected, refined, and filtered in those stages correspondingly. Another method SMA-Net was jointly proposed with the dataset MLT_S in \cite{6}. They adopted the encoder-decoder structure from PSPNet \cite{60}, and created a new multiscale attention module for accurate text segmentation.

2.3. Text Dataset

Spotlight datasets motivate researchers to invent effective methods to tackle computer vision problems. For example, the MNIST dataset of handwritten digits \cite{30} illustrated the effectiveness of a set of classical algorithms, e.g., KNN \cite{1}, PCA \cite{42}, SVM \cite{13}, etc. In recent years, the huge success of deep learning inspires researchers to create more challenging datasets to push forward the vision research front. Many text datasets are created for OCR purpose like CUTE80 \cite{43}, MSRA-TD500 \cite{58}, ICDARs \cite{25, 26, 39}, COCO-Text \cite{49}, and Total-Text \cite{10}, which are scene text datasets with word-boundary bounding boxes. Other datasets such as Synth90K \cite{24} and SynthText \cite{19} are synthetic text dataset for recognition and detection. Among these dataset, ICDAR13 \cite{26} and Total-Text \cite{10} provide pixel-level labels for text segmentation. Recently, Bonechi et al. introduced segmentation labels to COCO-Text and ICDAR17 MLT, forming up two new text segmentation datasets COCO_TS \cite{5} and MLT_S \cite{6}. In general, ICDAR13 and Total-Text are relatively smaller sets, and COCO_TS and MLT_S are of larger scale but their labeling quality is not precise.

3. Text Refinement Network

We propose a new approach, namely Text Refinement Network (TexRNet), specifically targets text segmentation. Since text segmentation is intrinsically similar to modern semantic segmentation, the related state-of-the-art methods can be leveraged to provide the base for our proposed TexRNet. Figure 3 overviews the pipeline of TexRNet, which consists of two components: 1) a backbone and 2) the key features pooling and attention module that refines the backbone for the text-domain. The design of the latter module is inspired by the uniqueness of text segmentation, and the principles will be discussed in Section 3.1. The network structure and corresponding loss functions will be detailed in Sections 3.2 and 3.3, respectively.

3.1. Design Principle

Multiple unique challenges distinguish text segmentation from modern semantic segmentation, thus motivating specific designs for text segmentation. In semantic segmentation, common objects, e.g., trees, sky, cars, etc., tend to share texture across different scenes. However, in text segmentation, the text texture may be extremely diverse across different words, although it could be homogeneous inside each word. To accommodate larger texture diversity, TexRNet dynamically activates low-confidence areas according to their global similarity to high-confidence regions, i.e., the yellow block in Figure 3, which aims to adaptively find similar textures in the same scene while relaxing the model from “remembering” those diverse textures.

Another challenge of text segmentation is the arbitrarily scaled text. The commonly adopted convolutional layers in semantic segmentation would limit the receptive field, reducing adaptiveness to diverse scale and aspect ratio. To achieve higher adaptiveness to scale, we adopt the popular non-local concept \cite{51, 48}. We use dot product and softmax to enforce attention on similar texture across the entire image.

3.2. Network Structure

As aforementioned, the backbone can employ an arbitrary semantic segmentation network. Here, we choose two representative works, i.e., ResNet101-DeepLabV3+ \cite{8} and HRNetV2-W48 \cite{50}, because they are the milestone and state-of-the-art in semantic segmentation, respectively. The rest of this section will focus on the new designs of TexRNet, i.e., the yellow block in Figure 3, which is the key to boosting text segmentation performance.

Assume an input image \( x \in \mathbb{R}^{H \times W \times 3} \), where \( H \) and \( W \) denote image height and width, respectively. The feature map extracted from the backbone is \( x_f \). The remainder of the proposed TexRNet could be described in the following three sequential components.

**Initial Prediction:** Similar to most traditional segmentation models, the feature map \( x_f \) is mapped to the semantic map \( x_{sem} \) through a convolutional layer (the kernel size is \( 1 \times 1 \)) with bias. After the softmax layer, \( x_{sem} \) becomes the initial segmentation prediction \( x'_{sem} \), which can be supervised by ground truth labels as the following.

\[
\mathcal{L}_{sem} = \text{CrossEntropy}(x'_{sem}, x_{gt}),
\]

where \( x'_{sem} = \text{Softmax}(x_{sem}) \), and \( x_{gt} \) indicates the ground truth label.
denotes the number of classes, and \( n \) in a single channel.

Because text does not have a standard texture that can be learned during training, the network must determine that text texture during inference. Specifically, the network should revise low-confidence regions if they share similar texture with high-confidence regions of the same class. To achieve this goal, we need to pool the key feature vector from high-confidence regions for each class \( i \in C \) to summarize the global visual property of that class. In our case, \(|C| = 2\), corresponding to text and background. More specifically, we conduct a modified cosine-similarity on the initial prediction \( x'_{\text{sem}} \) and use its output as new biases to transform \( x'_{\text{sem}} \) into \( \tilde{x}'_{\text{sem}} \) which is the weight map for key pooling. The cosine-similarity is written in Eq. 2, assuming \( x'_{\text{sem}} \in \mathbb{R}^{c \times n} \), where \( c = |C| \) denotes the number of classes, and \( n \) is the number of pixels in a single channel.

\[
\text{CosSim}(x'_{\text{sem}}) = X \in \mathbb{R}^{c \times c},
\]

\[
X_{ij} = \begin{cases} 
  x_i x_j^T / \|x_i\| \cdot \|x_j\|, & i \neq j \\
  0, & i = j 
\end{cases} 
\] (2)

\[
x_i = x'_{\text{sem}}^{(i)} \in \mathbb{R}^{1 \times n}, i = 1, \cdots, c,
\]

where \( \text{CosSim}(\cdot) \) denotes the modified cosine-similarity function, and \( x'_{\text{sem}}^{(i)} \) denotes the \( i \)th channel of \( x'_{\text{sem}} \), i.e., the predicted score map on class \( i \). From our empirical study, the cosine-similarity value \( X_{ij} \) indicates the ambiguity between prediction on classes \( i \) and \( j \). For example, when \( X_{ij} \) is close to 1, pixels are activated similarly in both \( x'_{\text{sem}} \) and \( x'_{\text{sem}}^{(j)} \) and thus cannot be trusted. Therefore, we use zero bias on class \( i \) and use biases in proportional to \( X_{ij} \) on class \( j \neq i \) equivalent to decrease the confidence scores on class \( i \). Those regions remains high-activated in class \( i \) are then used to the final prediction through a deep model. The final output \( x_{rfn} \) is supervised by the ground truth as shown in the following.

\[
\mathcal{L}_{rfn} = \text{CrossEntropy}(x_{rfn}, x_{gt}).
\] (5)
3.3. Trimap Loss and Glyph Discriminator

Since human vision is sensitive to text boundaries, segmentation accuracy along the text boundary is of central importance. In addition, text typically has a relatively high contrast between the foreground and background to make it more readable. Therefore, a loss function that focuses on the boundary would further improve the precision of text segmentation. Inspired by [23], we proposed the trimap loss as expressed as follows,

\[
\mathcal{L}_{tri} = \text{WCE}(x_{rfn}, x_{gt}, w_{tri}),
\]

\[
\text{WCE}(x, y, w) = -\frac{\sum_{j=1}^{n} w_j \sum_{i=1}^{c} x_{i,j} \log(y_{i,j})}{\sum_{j=1}^{n} w_j}
\]

where \(w_{tri}\) is the binary map with value 1 on text boundaries and 0 elsewhere, and \(\text{WCE}(x, y, w)\) is cross-entropy between \(x\) and \(y\) weighted by the spatial map \(w\).

Another unique attribute of text is its readable nature, i.e., the segments of glyphs should be perceptually recognizable. Given that the partial segmentation of a glyph diminishes its readability, we train a glyph discriminator to improve text segments’ readability. It is worth noting that the glyph discriminator also improves the evaluation score, as shown in the evaluation. More specifically, we pre-train a classifier for character recognition given the ground-truth character bounding boxes in the training set (the proposed dataset TextSeg provides these annotations). In our case, there are 37 classes, i.e., 26 letters, 10 digits, and misc. During the training of TexRNet, the pre-trained classifier is frozen and applied to the initial prediction \(x'_{sem}\), serving as the glyph discriminator. As illustrated in Figure 3, \(x'_{sem}\) is cropped into patches according to the character locations and then fed into the discriminator to obtain the discriminator loss \(\mathcal{L}_{dis}\), which indicates whether and how these patches are recognizable.

Unlike \(\mathcal{L}_{tri}\) that operates on \(x_{rfn}\), the glyph discriminator is applied on the initial prediction \(x'_{sem}\) for mainly two reasons: 1) \(\mathcal{L}_{tri}\) focuses on boundary accuracy while \(\mathcal{L}_{dis}\) focuses on the body structure of the text, which “distracts” each other if they are applied on the same prediction map. Our empirical studies also show that the improvements from \(\mathcal{L}_{tri}\) and \(\mathcal{L}_{dis}\) would be diminished if they work together on the same output, which aligns with our analysis. 2) \(\mathcal{L}_{tri}\) can directly impact the performance, so it oversees the model’s final output \(x_{rfn}\), while \(\mathcal{L}_{dis}\) reinforces the deep perception on text thus it can be placed on earlier layers. Above all, the final loss of TexRNet will be

\[
\mathcal{L} = \mathcal{L}_{sem} + \alpha \mathcal{L}_{rfn} + \beta \mathcal{L}_{tri} + \gamma \mathcal{L}_{dis},
\]

where \(\alpha, \beta, \) and \(\gamma\) are weights from 0 to 1. In the following experiments, \(\alpha = 0.5\), \(\beta = 0.5\), and \(\gamma = 0.1\). We select these loss weights in the way that the weight sums on two branches are roughly balanced (i.e. 0.5 + 0.5 ≈ 1 + 0.1).

4. The New Dataset TextSeg

As text in the real world is extremely diverse, to bridge text segmentation to the real world and accommodate the rapid advances of the text vision research, we propose a new dataset TextSeg, a multi-purpose text dataset focused on but not limited to segmentation.

4.1. Image Collection

The 4,024 images in TextSeg are collected from posters, greeting cards, covers, logos, road signs, billboards, digital designs, handwriting, etc. The diverse image sources could be roughly divided into two text types: 1) scene text, e.g., road signs and billboards, and 2) design text, e.g., artistic text on poster designs. Figure 2 shows examples of the two types. Existing text-related datasets tend to focus on scene text, while TextSeg balances the two text types to achieve a more real-world and diverse dataset. In addition, rather than focusing on text lines, the proposed TextSeg includes a large amount of stylish text. Sharing the language setting from those representative text segmentation datasets, the proposed TextSeg mainly focuses on English (i.e., case-sensitive alphabet, numbers, and punctuation).

4.2. Annotations

TextSeg provides more comprehensive annotations as compared to existing datasets. More specifically, TextSeg has annotated the smallest quadrilateral, pixel-level masks, and transcription for every single word and character. Besides, text effects, e.g., shadow, 3D, halo, etc., are annotated in TextSeg, which distinguishes text from traditional objects and significantly affects text segmentation. To the best of our knowledge, the proposed TextSeg is the only dataset with such comprehensive annotation for text segmentation.

Smallest Quadrilaterals are annotated to tightly bound words, characters, and punctuation. These quadrilaterals are recorded in the image coordinate (i.e., top-left origin, \(x\) axis is horizontally right, and \(y\) axis is vertically down), and the vertices are ordered clockwise starting from the top-left corner in the natural reading direction. A smallest quadrilateral tightly bounds a word or character, as shown in Figure 1. In certain cases like blurry text or long strokes, the quadrilaterals would cover the text’s core area by ignoring the ambiguous boundary or decorative strokes.

Pixel-level Masks consist of word masks, character masks, and word-effect masks. The word mask is a subset of the word-effect mask since the word mask labels the word surface without the effects like shadow and decoration, while the effect mask covers both word and effects. Similar to word masks, the character masks label character surfaces without those effects. Borrowing the concept from modern segmentation, word masks enable semantic segmentation, and character masks allow instance segmentation. For character masks, the most challenging cases are
Table 1: Statistical comparison between TextSeg and other datasets for text segmentation. The “–” marker indicates absence of the corresponding annotation in a dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Images</th>
<th>Approx. Image Size</th>
<th>Text Type</th>
<th># Bounding Polygons</th>
<th>Word-level Masks</th>
<th># Character Masks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scene Text Segmentation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICDAR13 FST [26]</td>
<td>462</td>
<td>1000 × 700</td>
<td>Scene</td>
<td>1,944</td>
<td>Word</td>
<td>4,786</td>
</tr>
<tr>
<td>COCO_Ts† [5]</td>
<td>14,690</td>
<td>600 × 480</td>
<td>Scene</td>
<td>139,034</td>
<td>Word</td>
<td>–</td>
</tr>
<tr>
<td>MLT_S† [6]</td>
<td>6,896</td>
<td>1800 × 1400</td>
<td>Scene</td>
<td>30,691</td>
<td>Word</td>
<td>–</td>
</tr>
<tr>
<td>Total-Text [10]</td>
<td>1,555</td>
<td>800 × 700</td>
<td>Scene</td>
<td>9,330</td>
<td>Word</td>
<td>–</td>
</tr>
<tr>
<td><strong>TextSeg (Ours)</strong></td>
<td>4,024</td>
<td>1000 × 800</td>
<td>Scene + Design</td>
<td>15,691</td>
<td>Word, Word-Effect</td>
<td>72,254</td>
</tr>
</tbody>
</table>

The 14,690 images in COCO_Ts is a subset of the totally 53,686 images in COCO-Text [19]. Similarly, the 6,898 images in MLT_S is a subset of the 10,000 images in ICDAR17 MLT [39]. Thus, their word bounding polygons can be directly extracted from their parent datasets.

Figure 4: Statistics of TextSeg. (a) Number of images with different numbers of words and characters. (b) Text coverage ratio against the image. (c) Character frequency of the whole dataset.

4.3. Statistical Analysis

Statistical comparison between TextSeg and four representative text segmentation datasets is listed in Table 1, i.e., ICDAR13 FST [26], MLT_S [6], COCO_Ts [5], and Total-Text [10]. In general, TextSeg has more diverse text types and all types of annotations. Another dataset that provides character-level annotations is ICDAR13 FST, but its size is far smaller than other datasets. COCO_Ts and MLT_S are relatively large, but they lack character-level annotations and mainly focus on scene text. The Total-Text was proposed with similar scope to those existing datasets.

The 4,024 images in TextSeg are split into training, validation, and testing sets with 2,646, 340, and 1,038 images, respectively. In TextSeg and all its splits, the ratio between the number of scene text and design text is roughly 1:1. Figure 4a counts the number of images with different numbers of words and characters, where 12-16 characters and 2-4 words per image is the majority. Figure 4b shows the distribution of the text coverage ratio, where the blue line is set up for word masks and the orange line is for word-effect masks. The rightward shifting from blue to orange indicates the coverage increment due to the word-effect. Finally, Figure 4c displays the character frequency in TextSeg, which roughly aligns with that of English corpus.

4.4. Qualitative Comparison

Figure 5 shows qualitative comparison between TextSeg, ICDAR13 FST, COCO_Ts, MLT_S, and Total-Text. ICDAR13 FST has many box-shape masks (considered as ignored characters), which is not a common case in the proposed TextSeg. Other datasets, i.e., COCO_Ts, MLT_S, and Total-Text, have only word masks. Note that COCO_Ts and MLT_S introduce a large number of ignored areas, especially along text boundaries, which would hinder models from precisely predicting text boundaries. Those boundary-ignored annotations are caused by automatic labeling using weekly supervised models. Similar to TextSeg, Total-Text is labeled manually, but it is of a much smaller size than ours and lacks annotations of characters and text effects.

5. Experimental Evaluation

To demonstrate the effectiveness of the proposed TextR-Net, it will be compared to the state-of-the-art methods DeeplabV3+ [8] and HRNet-W48 [50] on five datasets, i.e., ICDAR13 FST [26], COCO_Ts [5], MLT_S [6], Total-Text [10], and the proposed TextSeg.

5.1. Experiment Setup

Each model in comparison will be re-trained on each of the aforementioned text segmentation datasets. The models are initialized by ImageNet pretrains and then trained on 4 GPUs in parallel using SGD with weight decay of 5e−4 for 20,500 iterations. The first 500 iterations are linear warm-ups [18], and the rest iterations use poly decayed learning rates starting from 0.01 [8]. Note that 5,500 iterations are performed on ICDAR13 FST due to its small size as shown in Table 1. For TextSeg, our model train and evaluate using word masks as foreground instead of the word-effect masks. For the data augmentation, we randomly scale the short side
of the input images from 513 to 1025 and randomly crop a 513×513 patch as input in training.

The glyph discriminator in TexRNet adopts a ResNet50 classifier [22], which is trained on character patches from TextSeg training and validation sets. It achieves the classification accuracy of 93.38% on the TextSeg testing set. Since only the proposed TextSeg and ICDAR13 FST provide character bounding boxes, the glyph discriminator is only applied on these two datasets and disabled on COCO_TS, MLT_S, and Total-Text.

We evaluate our models using multi-scale no-flip ensemble 8 scales from 0.75x to 2.5x of a standard 513 short side image. To align with modern segmentation tasks, we use foreground Intersection-over-Union (fgIoU) as our major metric. Also, the typical F-score measurement on foreground pixels is provided in the same fashion as [11, 26]. The foreground here indicates the text region in both prediction and ground truth.

5.2. Model Performance

This section compares TexRNet to other text and semantic segmentation methods. To demonstrate the effectiveness of TexRNet, the comparison is conducted on five datasets including our TextSeg. As previously claimed, we adopt DeeplabV3+ [8] and HRNetV2-W48 [50] as our backbone and baseline. We also compare with the SOTA semantic segmentation model: HRNetV2-W48 + Object-Contextual Representations (OCR) [59]. The PSPNet and SMANet results are from [5, 6] in which their models were trained on ICDAR13 FST and Total-Text augmented with SynthText [19]. Tables 2 shows the overall results. As the table shows, our proposed TexRNet outperforms other methods on all datasets.

5.3. Ablation Studies

This section performs ablation studies on the key pooling and attention (the yellow block in Figure 3), trimap loss, and glyph discriminator in the proposed TexRNet. In this experiment, DeeplabV3+ is adopted as the backbone, and the models are trained and evaluated on TextSeg. Starting from the base version of TexRNet, the key pooling and attention (Att.), trimap loss ($\mathcal{L}_{\text{tri}}$), and glyph discriminator ($\mathcal{L}_{\text{dis}}$) are added incrementally as shown in Table 3, where the fgIoU and F-score are reported, presenting a consistently increasing trend. The final TexRNet achieves the best performance, around 2% increase in fgIoU as compared to DeeplabV3+.

An interesting observation is that TexRNet (final) have exactly the same number of parameters as TexRNet (base), but the part between them contributes the most improvement. To further investigate whether the performance increase comes from parameter increase, we compared TexRNet with HRNetV2-W48+OCR and other models in Figure 6. We discover that TexRNet achieves higher accuracy with less parameters as compared to HRNetV2-W48+OCR.

Table 2: Performance comparison between TexRNet and other models on TextSeg and other representative text segmentation datasets. The bold numbers indicate the best results.

<table>
<thead>
<tr>
<th>Method</th>
<th>TextSeg (Ours)</th>
<th>ICDAR13 FST</th>
<th>COCO_TS</th>
<th>MLT_S</th>
<th>Total-Text</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fgIoU</td>
<td>F-score</td>
<td>fgIoU</td>
<td>F-score</td>
<td>fgIoU</td>
</tr>
<tr>
<td>PSPNet† [60, 5]</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.797</td>
<td>–</td>
</tr>
<tr>
<td>SMANet† [6]</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.785</td>
<td>–</td>
</tr>
<tr>
<td>DeeplabV3+ [8]</td>
<td>84.07</td>
<td>0.914</td>
<td>69.27</td>
<td>0.802</td>
<td>72.07</td>
</tr>
<tr>
<td>HRNetV2-W48 + OCR [59]</td>
<td>85.03</td>
<td>0.914</td>
<td>70.98</td>
<td>0.822</td>
<td>68.93</td>
</tr>
<tr>
<td>Ours: TexRNet + DeeplabV3+</td>
<td>86.06</td>
<td>0.921</td>
<td>72.16</td>
<td>0.835</td>
<td>73.98</td>
</tr>
<tr>
<td>Ours: TexRNet + HRNetV2-W48</td>
<td>86.84</td>
<td>0.924</td>
<td>73.38</td>
<td>0.850</td>
<td>72.39</td>
</tr>
</tbody>
</table>

† In [5, 6], the author augmented the original training dataset with SynthText [19] in both ICDAR13 FST and Total-Text experiments.
<table>
<thead>
<tr>
<th>Method</th>
<th>Att.</th>
<th>$L_{tri}$</th>
<th>$L_{dis}$</th>
<th>fgIoU</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeeplabV3+</td>
<td></td>
<td></td>
<td></td>
<td>84.07</td>
<td>0.914</td>
</tr>
<tr>
<td>TexRNet (base)</td>
<td>✓</td>
<td></td>
<td></td>
<td>84.86</td>
<td>0.917</td>
</tr>
<tr>
<td>TexRNet</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>85.36</td>
<td>0.919</td>
</tr>
<tr>
<td>TexRNet</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>85.55</td>
<td>0.921</td>
</tr>
<tr>
<td>TexRNet (final)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>86.06</td>
<td>0.921</td>
</tr>
</tbody>
</table>

Table 3: Ablation studies of TexRNet on TextSeg. All models are training on TextSeg train and validation sets, and all TexRNet use DeeplabV3+ as backbone. The column “Att.” represents whether attention layers are included. Similarly, columns “$L_{tri}$” and “$L_{dis}$” indicate whether the trimap loss and glyph discriminator are used.

Figure 6: Comparison of different methods in the number of parameter vs. text segmentation performance in fgIoU, demonstrating the effectiveness of our design in TexRNet.

5.4. Downstream applications

This section gives prospects of TexRNet and TextSeg dataset, especially in driving downstream applications.

**Text Removal** is a practical problem in photo and video editing, and it is also an application with high industrial demand. For example, media service providers frequently need to erase brands from their videos to avoid legal issues. Since this task is a hole filling problem, Deep Image Prior [47] is employed, and different types of text masks are provided to compare the performance of text removal. Typically, word or character bounding boxes are standard text masks because they are easy to get from existing text detection methods. By contrast, the proposed TexRNet provides much more accurate text masks. Figure 7 compares the results using these three types of text masks, i.e., text segmentation mask, character bounding polygon, and word bounding polygon. Obviously, finer mask yields to better performance.

**Text Style Transfer** is another popular task for both research and industry. Mostly, text style transfer relies on accurate text masks. In this experiment, we use Shape-Matching GAN [56] as our downstream method, which requires text masks as an input. In their paper, all demo images are generated using ground truth text masks, which may be impractical in real-world applications. Therefore, we extend TexRNet with Shape-Matching GAN to achieve scene text style transfer on an arbitrary text image. A few examples are visualized in Figure 8, and more examples can be found in the supplementary.

6. Conclusions

We introduce a novel text segmentation dataset TextSeg, which consists of 4,024 scene text and design text images with comprehensive annotations including word- and character-wise bounding polygons, masks, and transcriptions. We also propose a new and effective text segmentation method, TexRNet. We demonstrate that our model outperforms state-of-the-art semantic segmentation models on TextSeg and another four datasets. To support our idea that text segmentation has great potential in the industry, we introduce two downstream applications, i.e., text removal and text style transfer, to show promising results using text segmentation masks from TexRNet. In conclusion, text segmentation is a critical task. We hope that our new dataset and method would become the cornerstone for future text segmentation research.
References


