RecycleNet: An Overlapped Text Instance Recovery Approach

Yiqing Hu*
Tencent YouTu Lab
hooverhu@tencent.com

Yan Zheng*
Tencent YouTu Lab
neoyzheng@tencent.com

Xinghua Jiang
Tencent YouTu Lab
clarkjiang@tencent.com

Hao Liu
Tencent YouTu Lab
ivanhliu@tencent.com

Deqiang Jiang
Tencent YouTu Lab
dqiangjiang@tencent.com

Yinsong Liu
Tencent YouTu Lab
jasonysliu@tencent.com

Bo Ren
Tencent YouTu Lab
timren@tencent.com

Rongrong Ji
Xiamen University
rrji@xmu.edu.cn

ABSTRACT

Text recognition is the key pillar for many real-world multimedia applications. Existing text recognition approaches focus on recognizing isolated instances, whose text fields are visually separated and have no interference with each other. Moreover, these approaches cannot handle overlapped instances that often appear in sheets like invoices, receipts and math exercises, where printed templates are generated beforehand and extra contents are added afterward on existing texts. In this paper, we aim to tackle this problem by proposing *RecycleNet*, which automatically extracts and reconstructs overlapped instances by fully recycling the intersecting pixels that used to be obstacles for recognition. RecycleNet parallels to existing recognition systems, and serves as a plug-and-play module to boost recognition performance with zero-effort. We also released an OverlapText-500 dataset, which helps to boost the design of better overlapped text recovery and recognition solutions.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; • **Applied computing** → *Document management and text processing*.

KEYWORDS

optical character recognition; semantic segmentation

ACM Reference Format:
Yiqing Hu, Yan Zheng, Xinghua Jiang, Hao Liu, Deqiang Jiang, Yinsong Liu, Bo Ren, and Rongrong Ji. 2021. RecycleNet: An Overlapped Text Instance Recovery Approach. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3481536

*Both authors contributed equally to this research.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3481536

1 INTRODUCTION

Text recognition aims to parse a natural image and extract character sequences based on recognizing visual contents. With the development of deep learning, the ability of text recognition has rapidly boosted, which facilitates many new applications, such as document understanding [38] and automatic exercise correction [8]. Owing to its huge potential, text recognition has attracted increasing attention in the multimedia and computer vision community. Recent advances [7, 11, 12, 16] have already yielded exciting progress. With cutting-edge model design and data augmentation, existing approaches have enabled accurate text recognition against rotated and fuzzy targets, while in the meantime making noises like color jitter no longer obstacles. Unfortunately, these approaches cannot handle overlapped texts that often appear in sheets like invoices, receipts and math exercises, where printed templates are generated beforehand, and extra contents are added afterward on existing texts, as shown in Fig. 1. These phenomena are a dilemma for real-world text recognition applications, which however are not solved yet. In this paper, we aim to tackle this problem by proposing *RecycleNet*, which automatically retrieves and renovates all overlapped text instances.

In order to recognize overlapped texts, an intuitive idea is to reinforce the recognition system. To achieve this goal, recognition models could be trained by pairs of images and concatenated labels of overlapped text instances. However, such a trial is less effective in practice. To explain, mainstream recognition approaches follow a sequence-to-sequence rule to translate a series of visual feature slices to a series of character sequences [29]. Thus each slice mainly stands for a single particular character or null. And such a setting unfortunately does not hold in the scene of overlapped text, where a feature slice may indeed represent multiple symbols, as shown in Fig. 2. We also turn on another pipeline, which focuses on a single instance while treating the rest as noises, then forces the model to ignore these noises. Experimental results foreclosed this possibility, as the noise is too strong for accurate recognition.

To avoid the recognition conundrum, another solution is to separate individual instances and then feed them to the recognition module sequentially. Intuitively, semantic segmentation [28, 37] or instance segmentation [5] can realize this goal, which assign pixel-level label-maps, namely masks, to outputted images, thus

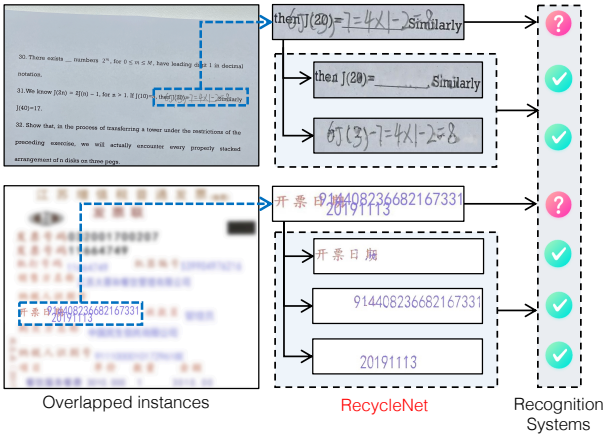

Figure 1: Overlapped texts in real-world scenes. Existing recognition approaches suffer in parsing these texts directly. RecycleNet recovers all overlapped instances by "recycling" the intersecting pixels that used to be obstacles for recognition. The easy inputs provided by RecycleNet facilitate following recognition systems.

making it possible to retrieve silhouette of text instances. However, they are still inadequate for overlapped instance separation. To explain, an overlapped pixel belongs to multiple instances but has only one label. Hence, although one instance will get it, the others could be lost. This phenomenon precludes text recognition, for many symbols will change to another after losing part of the strokes. For instance, number '7' will change to '1' after losing the upper horizontal stroke.

Essentially overlapped text instance recovery can be resorted to a multi-pixel multi-use problem. If overlapped pixels could be located and returned to related instances, these instances could be reconstructed without losing any strokes. In other words, these pixels should be "recycled". The proposed RecycleNet derives from this observation, and is further motivated by recent advances in image matting. The goal of image matting is to estimate per-pixel opacity of unknown regions, then filters out the valuable parts to complement foreground regions. With a similar goal but with a more complex situation, RecycleNet pinpoints all multi-use pixels and exploits them for complement. To realize this goal, RecycleNet is designed as an end-to-end trainable, two-stage network. The first stage focuses on text silhouette extraction, assigning a multi-hot label to both overlapped and non-overlapped pixels. Each hot indicates the possibility that it belongs to a particular text instance. The second stage focuses on overlapped pixel optimization, using the obvious strokes provided by the first stage as the guidance. Finally, RecycleNet outputs reconstructed instances, which are easy inputs for following recognition systems if they exist. For these systems, RecycleNet not only increases accuracy but also avoids complicated network design.

The contributions of this paper are summarized as follows:

- The proposed RecycleNet is the first overlapped text instance recovery approach. It parallels to existing text recognition and text spotting systems, and could be a plug-and-play module for them to achieve better performance.
- Experimental results show that RecycleNet boosts overlapped text recognition remarkably. And it is integrated into multiple real-world commercial on-line text recognition services, processing over 200, 000 images on average per day to tackle overlapped texts.
- There is no comprehensive benchmark publicly available to evaluate overlapped text recovery and recognition. We handle this issue by establishing an OverlapText-500 dataset. This benchmark dataset can further inspire the design of better overlapped text recovery and recognition approaches.

2 RELATED WORKS

RecycleNet is designed as a front module for text recognition systems, and could also be plugged into text spotting systems, in which text detection and recognition are integrated. In this section, we briefly introduce these related researches.

2.1 Text detection

Texts on the input image could be located by both text detection and segmentation solutions. Motivated by mainstream objection detection frameworks [15, 26, 27], text detection approaches are designed and further customized according to intrinsic features of text instances [13, 30]. Beyond regular text detection, recent advances focus on more challenging oriented text and twisted text. Features of contour [31], center [47], corner [19] and even relationship between individual characters [45] are utilized to guide network design. In addition, such visual features could be directly encoded into the network to represent texts in horizontal, oriented or curved forms [18, 32]. Compared with the detection counterparts, segmentation is characterized by fine-grained outputs. The targets could be instance-level [5, 35] or even pixel-level with corresponding post-processing steps [2]. For both detection and segmentation, their common goal is to robustly extract all text instances against heterogeneous noises, such as the noise caused by overlapped text. With careful model design and data augmentation, they could yield their own goal, but however, still left the overlapping noises unchanged on extracted texts. As a consequence, this problem is passed to the following recognition systems, if they exist.

2.2 Text recognition and spotting

Existing text recognizers could be roughly divided into three types: 1) classification-based, 2) sequence-to-label-based and 3) sequence-to-sequence-based. Now sequence-to-sequence approaches are mainstream owing to their excellent performance in utilizing contexts. With a CNN-based feature extractor and a sequence generator, a series of visual features is translated to a series of characters. Sequence generators could be categorized into two types: Connectionist Temporal Classification (CTC) based [6, 29] and attention based [10, 39]. As fuzzy targets are hard to be recognized, linguistic knowledge is introduced and encoded into networks to correct suspicious recognition results [24, 34]. Besides, a number of designs [33, 40, 42, 43] have been proposed to better balance visual and contextual knowledge.

In addition to text recognition approaches, text spotting systems pack text detection and text recognition together. These systems

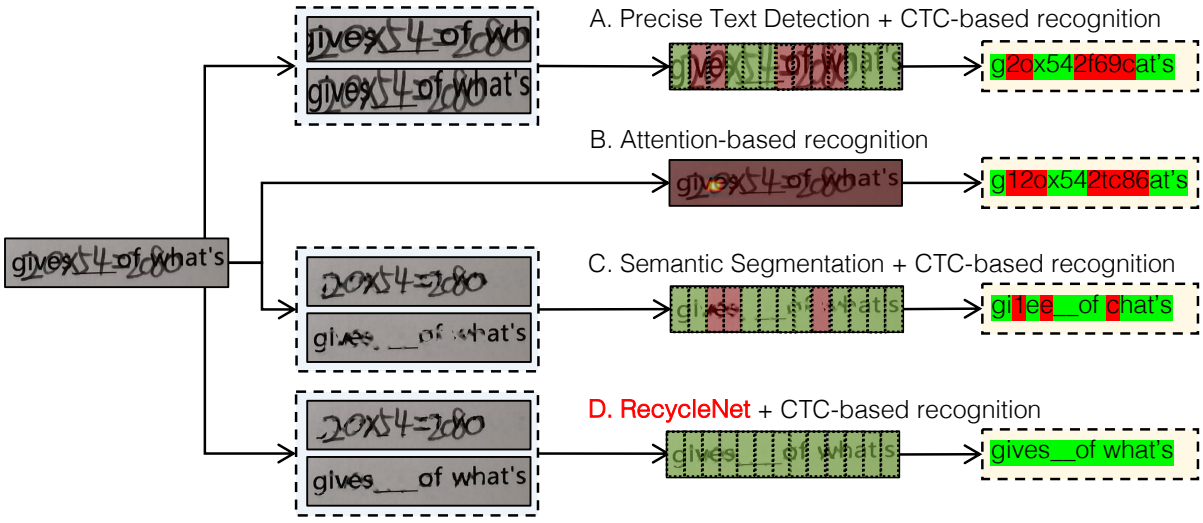

Figure 2: Several solutions to recognize overlapped texts. CTC-based approaches (A) require a robust single-line text detection module while attention-based approaches (B) do not, but they both cannot accurately resolve feature slices that represent multiple symbols. Segmentation approaches (C) could extract silhouette of texts, but there always exists result with obvious stroke loss. RecycleNet (D) recovers all instances with fine-grained strokes. Best viewed in color.

mainly consist of two types: 1) using separate detection and recognition modules [13] and 2) using a unified framework [11, 16]. The unified systems are expected to achieve better system efficiency and efficacy, as they enable feature fusion among multiple tasks. With character-level annotations, in [12], it harnesses feature provided by the detection module to reinforce the recognition module. Furthermore, to overcome the lack of character-level annotations, a weak-supervised approach [36] is designed to predict character-level result. Notice that existing recognition and spotting systems are designed to parse isolated texts. However, this does not always hold in real-world scenes, especially for sheets like invoices, receipts and math exercises, where extra contents are added unpredictably on existing printed templates. RecycleNet is designed to avoid this recognition conundrum, and it further provides easy inputs for existing recognition approaches.

3 APPROACH

RecycleNet consists of two sub-networks: a Silhouette Extraction Network (E-Net for short) and a Silhouette Perfection Network (P-Net for short). For real-world applications, a text recognition module should be appended; Furthermore, to build a text spotting system, a front text detection module is required, as depicted in Fig. 3. We focus on the design of RecycleNet in this paper.

3.1 Silhouette Extraction Network

Given an input image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$ with N text instances on it, The goal of RecycleNet is to extract all these instances with minimum stroke loss, which is crucial to a following recognition system. Notice that instances are not necessarily all overlapped; A common scene is that instances are printed or written orderly onto a sheet. That is, instance I_1 first appears on the background, then $I_2, ..., I_N$. Hence, the following instances intersect previous ones with probability. Each instance I is formed by a group of pixels $\{c\}$.

Different instances are characterized by a unique set of color, font, location, length, height and orientation. Among them, color is an apparent visual feature. Intuitively, we argue that if it is possible to separate instances when they have distinct colors. By specifying a fix cluster number, color clustering is conducted and the result is shown in the first column of Fig. 4. We see that the result is not satisfactory even with a tailored configuration. The insight is that in real-world cases, the feature distribution is not that distinct but varies continuously due to environmental noise and imaging quality. As a result, edge pixels of instances are confusable and hard to be correctly classified. Exploiting other features also faces similar dilemmas.

Recent advances in semantic segmentation [28, 46] and instance segmentation [5] show superior performance, and it seems viable to directly adopt them. Using $\mathcal{I}$ as a input, these approaches output a same-sized, pixel-level label map, *i.e.*, a segmentation mask. Each pixel has a corresponding label in this mask that indicates its original instance. This label is usually an integer or a one-hot label when referring to multiple kinds of target. However, these approaches are still inadequate in our target scene. An overlapped pixel belongs to multiple instances but only corresponds to a single of them. In other words, one instance will get it and the other could be lost. This phenomenon precludes a following text recognition system, for many symbols are ambiguous after losing part of strokes, as shown in the second column of Fig. 4.

As each overlapped pixel belongs to multiple instances, we consider it should obtain a multi-hot label rather than a one-hot label, thus it could be "recycled" by all needed instances. Following this assumption, the label map of $\{c\}$ is defined as a multi-hot map $\mathcal{P} \in \mathbb{R}^{H\times W\times N}$. $\mathcal{P}$ could be regarded as a set that contains $H \times W$ elements. Each element is a N-hot vector, thus $\mathcal{P} = \{(p_1, p_2, ..., p_N)\}$. $p_k \in [0, 1]$ is a floating number indicates the possibility that a pixel

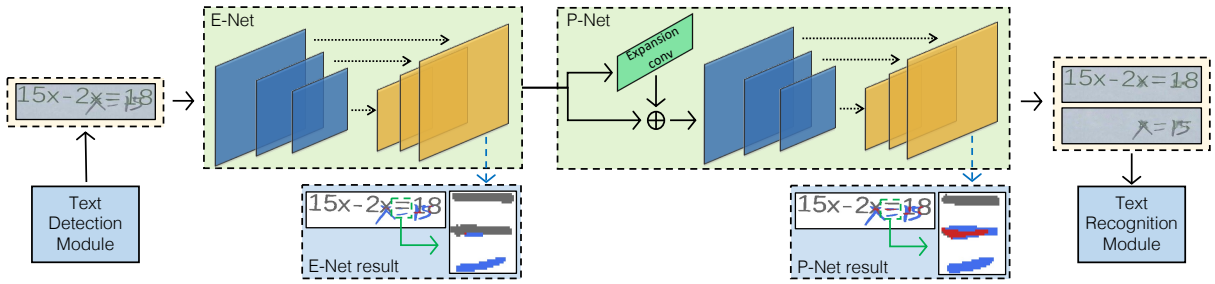

Figure 3: System overview. RecycleNet is a two-stage, end-to-end trainable network, in which 1) Silhouette Extraction Network (E-Net) focuses on text silhouette extraction, enables "recycle" of pixels by predicting multi-hot labels; 2) Silhouette Perfection Network (P-Net) focuses on the more challenging overlapped pixel optimization. Red pixels in enlarged parts of label maps indicate the predicted pixels to be "recycled" by multiple text instances.

belongs to instance k. Hence, E-Net predicts label map $\mathcal{P}$ by:

$$\mathcal{P} = f(\mathcal{I}) \tag{1}$$

Here $f(\cdot)$ is a fully convolutional network. After traversing all pixels, the ones with their k th label element $p_k > \beta$ are selected, here β is a constant. Such a group of pixels represents instance I_k. Similarly, all instances could be recovered in this way.

3.2 Silhouette Perfection Network

Compared with existing solutions, Silhouette Extraction Network (E-Net) already yields noticeable improvement in recovering overlapped instances. However, harsh real-world scenes pose strong challenges as well, lead to obvious bad-cases such as broken stroke or even failures in key stroke recovery. These bad-cases are still caused by the ambiguous regions generated by overlapping text strokes. To better solve these problems, we try to discover other clues for guidance.

We exploit the stroke, which is an intrinsic feature of text, to guide searching for missing pixels. Through experiments of E-Net, we observed that the overlapped strokes are hard to be correctly returned to corresponding instances; On the other hand, the non-overlapped strokes are clear and easy to be classified. This is not a surprising result, but we observed that the missing pixels are usually extensions of clear strokes, which E-Net could accurately recover with a high probability. For a particular text instance, if its strokes suffer from pixel loss, by simply expanding its current boundary, the newly intersecting pixels with other instances are suspicious to be the missing parts, as shown in Fig. 5. Based on this observation, a subsequent network named Silhouette Perfection Network (P-Net for short) is designed to realize this goal.

P-Net is formed by a customized expansion convolution $g(\cdot)$ and a following convolutional network $f(\cdot)$ same with E-Net, as shown in Fig. 3. $g(\cdot)$ derives from the dilation operation, which is one of the two basic operators in the area of mathematical morphology, the other being erosion. The most commonly usage of dilation is to exaggerate features in an image that would otherwise be missed, which perfectly fits our purpose. To distinguish $g(\cdot)$ from dilated convolution [41], we name it as expansion convolution.

P-Net utilizes the outputted label map $\mathcal{P} \in \mathbb{R}^{H\times W\times N}$ of E-Net as input. Herein each slice $P_i = \{p_i\} \in \mathbb{R}^{H\times W\times 1}$ stands for label map of instance i. We apply expansion convolution $g(\cdot)$ on each slice. As a consequence, we have:

$$P_i' = g(P_i) = \{p_i'\} \tag{2}$$

Referring to P_i' and P_i, we denote $\{c_i'\}$ and $\{c_i\}$ as the corresponding pixels with non-zero labels. These pixels constitute text instances. Hence, pixels generated by expansion is denoted by:

$$\{\Delta c_i\} = \{c_i'\} - \{c_i\} \tag{3}$$

Similarly, we denote pixels with non-zero labels of instance j as $\{c_j\}$. If $\{\Delta c_i\}$ overlaps $\{c_j\}$, instance j may steal missing pixels of instance i. In this case, we update the i th label of these overlapping pixels, for their possibility to be part of instance i arise. Otherwise, the expansion operation is invalid and no action will be taken. Assuming a pixel $c \in \{\Delta c_i\}$, we update its i th label component by:

$$p_i = \begin{cases} p_i' & \text{if } c \in \{\Delta c_i\} \cap \{c_j\}, \textbf{s.t.} \ \forall j \in [1, N], j \neq i \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Apply above operations on all slices, we could get an expanded label map $\mathcal{P}'$. It explicitly encodes clear stroke information of related instances, which guides the network to learn ambiguous pixels.

We concatenate $\mathcal{P}$ with $\mathcal{P}'$ by using operation $\oplus$. The concatenation result is utilized as input of a same convolutional network $f(\cdot)$. $f(\cdot)$ produces outputs of P-Net, which is yet another label map $\mathcal{J} \in \mathbb{R}^{H\times W\times N}$:

$$\mathcal{J} = f(\mathcal{P} \oplus \mathcal{P}') \tag{5}$$

Similar to post-processing steps in E-Net, all instances could be recovered by selecting eligible pixels according to $\mathcal{J}$.

3.3 Implementation Details

Network design: Similar to semantic segmentation approaches, we design RecycleNet to enforce pixel-wise supervision. We simply adopt U-Net [28] as the backbone of both E-Net and P-Net for its compact structure and adequate functionality. Recent backbone designs like [25] will no doubt improve RecycleNet's performance, and we leave it for future exploration. U-Net is essentially a fully convolutional network with a symmetrical structure. It includes a series of down-sampling layers followed by a series of up-sampling layers. Feature maps are concatenated between corresponding down-sampling layers and up-sampling layers for fusion. The default U-Net supports 16× down-sampling and up-sampling, and we enlarge the network with additional blocks to support 32×.

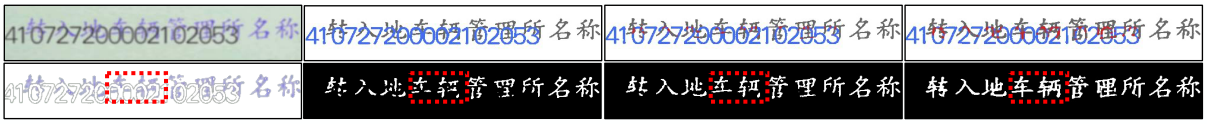

Figure 4: Performance of different overlapped text recovery approaches. From left to right, each column stands for 1) the original image on top, and an extracted instance from a customized two-class color clustering at bottom, 2) result of a segmentation network (a mask on top and an extracted instance at bottom), 3) result of Silhouette Extraction Network (E-Net) and 4) result of Silhouette Perfection Network (P-Net), respectively. P-Net recovers original strokes with high confidence. For masks in 3) and 4), red pixels indicate predicted pixels to be "recycled" by multiple texts while the others are not. Best viewed in color.

Experimental result in Section 4 demonstrates that a broader receptive field is necessary to capture instances' integrity. In addition, Coordinate Convolution [14] is added to the initial convolution block of both networks. It was shown to improve the generalization ability on coordinate-sensitive tasks. The maximum supported instance number N is set as 4, as a larger N is meaningless and rarely appears in real-world scenes. Threshold β is set as 0.5.

RecycleNet is a stacked architecture with E-Net in front and P-Net at back, which is inspired by recent researches of progressive refinement [1, 20, 22]. Specifically, P-Net includes a customized expansion operation (Section 3.2). Applying this operation equals to going through a convolutional layer, whose kernel is frozen with size and weight set fixed in advance. We conduct extensive experiments to search this kernel, and find that a kernel with 1) 5×5 window and 2) Gaussian weight decay from the kernel center performs best. The Gaussian kernel is with expectation $\mu = 0$ and standard deviation $\sigma = 1.1$. Intermediate prediction $\mathcal{P}$ passes through this layer and with its expanded counterpart $\mathcal{P}'$ generated. These two are concatenated together as input of the following convolution blocks.

Training strategy: The loss functions of E-Net and P-Net have a slight difference on account of their different emphases. E-Net comprehensively extracts text silhouette and assigns multi-hot labels to pixels, while P-Net focuses on recovering ambiguous pixels. For E-Net, we exploit Binary Cross Entropy (BCE) loss to calculate its loss L_s, using $\mathcal{P}$ as input. The same loss function is utilized in P-Net, but its loss L_p consists of two parts, namely the loss of non-overlapped pixels L_{no} and the loss of overlapped pixels L_o. As

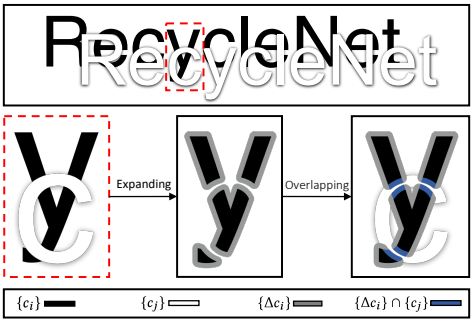

Figure 5: The expansion operation in P-Net. Symbol "c" overlaps "y" and takes away partial pixels of "y". After expanding current boundary of "y", the newly generated pixels that overlapped with "c" are suspicious to be missing parts of "y".

L_o is the principle goal of P-Net, we amplify its weight by λ. Hence, the loss of P-Net $L_p = L_{no} + \lambda L_o$. λ is set to 2 experimentally, and is used throughout the later experiment section.

The training process has two phases. In the first phase, E-Net and P-Net are trained separately for initialization, each for 100 epochs. The SGD optimizer with learning rate 0.01 is chosen for optimization. The learning rate halves after each 20 epochs. The training batch size is set as 80. In the second phase, these two sub-networks are trained jointly for another 50 epochs to improve overall performance. We do not rely on any pre-trained models. Basing on Pytorch [23], we implement all benchmarks on a regular platform with one Nvidia V100 GPU and 64GB memory.

4 EXPERIMENTS

4.1 Datasets and Settings

Training set: The training set includes two kinds of annotations, namely 1) separated masks of text instances and their overlapped parts, and 2) character sequence annotation of all appeared text instances. The label of text annotation comprises Arabic numbers, operation symbols (*e.g.*, $\times$), uppercase/lowercase English letters and 3,000 regular Chinese characters. To satisfy real-world usages, annotating a large amount of this complex training set manually is infeasible. As a consequence, we developed an image synthesis engine. It synthesizes a total number of 1 million images according to the previously mentioned annotation format. Synthetic images are with mean resolution 100×500, meanwhile $1 \sim 4$ text instances with decided masks randomly fall on each image. With these masks, the multi-hot label map of all pixels could be calculated according to the intersecting status of masks. For each text instance, our engine randomly distributes different fonts, colors, locations, lengths and heights to it. Number of orientated instances is controlled to 1 or 0 to approximate real-world scenes. In addition, color jitters and shadows are added randomly to mimic environmental noises.

OverlapText-500 testing set: The testing images come from two principal real-world scenarios: financial documents and math exercises. word / numeric overlapping usually occurs in the former and numerical / numerical overlapping often appears in the latter. In these two scenarios, templates are generated beforehand and extra contents are occasionally added afterward on existing printed texts. This testing set could be used to evaluate both RecycleNet and recognition systems. Due to its expensive cost, we annotated only 500 images for testing and prepared to have it released. The annotation format is same to that of the training set.

Criteria: we adopt two metrics for evaluation, one for text instance recovery (RecycleNet) and the other for text recognition and text

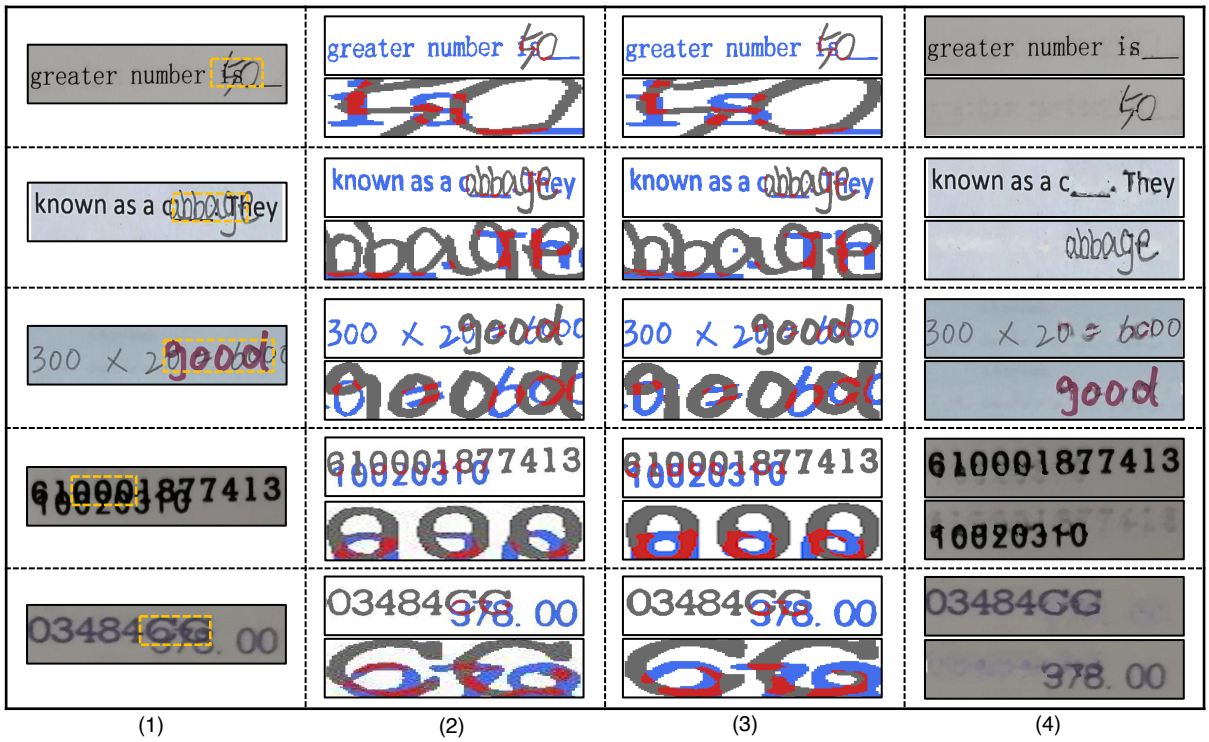

Figure 6: Performance of RecycleNet in real-world scenes. From left to right, each column stands for (1) original images with highlighted overlapping region, (2) ground-truth label-maps with focus on the marked region, (3) predicted label-maps of RecycleNet with focus on the marked region and (4) final outputs of RecycleNet, respectively. In label-maps, red pixels are pixels of overlapped strokes while the others are non-overlapped. Best viewed in color.

spotting. To comprehensively evaluate RecycleNet, five measures including (1) recall, (2) precision, (3) Mean Absolute Error (MAE), (4) Mean Intersection over Union (MIoU) and (5) S-measure [4] are used. The latter experiment section utilizes MIoU as the major measurement tool unless otherwise stated. For evaluating text recognition and text spotting performance, we select the evaluation protocols of ICDAR 2015 [9]. As overlapped texts are unpredicted character sequences, we choose the general "End-to-End" protocol for it works without a contextualized lexicon.

4.2 Ablation: Silhouette Extraction Network

Performance of Silhouette Extraction Network (E-Net) is influenced by multiple factors. We traverse several important of them for ablation. Besides, we also test E-Net in multiple challenging scenes to show its robustness.

Label dimension: Extending the pixel's label from one-hot label to multi-hot label facilitates extraction of text instances. To show its impact, we downgrade E-Net to a common U-Net with each pixel one-hot labeled, thus a pixel only belongs to a single text instance. Experimental result (Row 3 and Row 4 in Table 1) demonstrates that this operation leads to an obvious MIoU decrement from 78.14% to 73.60%.

Input format: The default network input is an $\mathbb{R}^{H\times W\times 3}$ colored image. To show impact of a colored input, all inputs are converted to $\mathbb{R}^{H\times W\times 1}$, and the network exploits these grayscale images as input. This operation (Row 3 and Row 5 in Table 1) decreases MIoU, which shows that color is still an important feature for instance extraction. Notice that a colored input could not be guaranteed in real-world applications. As we leverage the $\mathbb{R}^{H\times W\times 3}$ format as default, all inputs are transformed to this default format. RecycleNet is still robust when working on most grayscale inputs.

Text locations: Texts may appear at arbitrary locations on the input image. We consider location of instances is a useful clue for instance extraction. Hence, we apply Coordinate Convolution [14] in our network since this operation was shown to improve the generalization ability on coordinate-sensitive tasks. Experimental result (Row 3 and Row 6 in Table 1) demonstrates that removing Coordinate Convolution leads to a slight decrease of MIoU from 78.14% to 77.28%.

Receptive field: Instance integrity, *i.e.*, relation of pixels within a text instance, is another important factor for extraction. The default down-sample rate and up-sample rate of U-Net are 16×. We enlarge the receptive field to ensure the instance integrity could be better captured. Basing on this observation, we set the down-sample rate / up-sample rate to 32× by appending additional convolutional blocks. Experimental result (Row 3 and Row 7 in Table 1) demonstrates that a smaller receptive field also leads to an obvious decrease of MIoU from 78.14% to 76.54%.

Instance overlapping percentile: It is obvious that a more severe overlapping increases the difficulty of extraction. In this evaluation task, we evaluate E-Net's performance under different overlapping percentiles of text instances. To realize this goal, we carefully select

Table 1: Ablation study of Silhouette Extraction Network (E-Net).

	Text region					Overlapped regions in text region				
Method	Recall	Precision	MAE	MIoU	S-measure	Recall	Precision	MAE	MIoU	S-measure
E-Net	86.76	88.48	1.46	78.14	93.91	70.88	72.84	0.46	56.07	87.24
- Label dimension	80.96	89.01	1.73	73.60	91.88	-	-	-	-	-
- Colored input	85.55	87.28	1.60	76.07	93.22	63.70	66.34	0.57	48.14	84.47
- CoordConv [14]	85.93	88.48	1.50	77.28	93.66	65.76	70.45	0.51	51.54	85.43
- Reception Field	84.72	88.80	1.55	76.54	93.41	66.89	70.38	0.51	52.20	85.69
Overlapped percentile ($N = 2, \zeta = 15\%$)	89.23	85.36	1.26	77.38	93.57	73.31	57.54	0.29	47.57	84.66
Overlapped percentile ($N = 2, \zeta = 50\%$)	89.30	84.20	1.26	76.49	93.32	72.29	55.16	0.35	45.53	84.18
Overlapped percentile ($N = 2, \zeta = 90\%$)	73.97	72.74	2.59	57.91	86.82	63.04	46.25	0.96	36.38	80.29
Overlapped items ($N = 2, \zeta = 75\%$)	81.44	80.99	1.77	68.37	90.33	64.52	52.71	0.75	40.87	81.14
Overlapped items ($N = 3, \zeta = 75\%$)	55.39	59.74	4.43	40.33	71.47	38.60	43.32	1.71	25.65	62.57
Overlapped items ($N = 4, \zeta = 75\%$)	50.60	45.07	4.50	31.30	66.38	31.10	26.20	1.68	16.58	57.34

three subsets from our OverlapText-500 dataset, each includes 50 images. Each image in the set contains two instances, with a vertical overlapping percentile ranges in $[\zeta-10\%, \zeta+10\%]$, here ζ is a median threshold. Experimental results (Row 3 and Row 8∼10 in Table 1) depict that MIoU decreases with the increment of overlapping percentile. Notice that when the overlapping percentile reaches $\zeta = 90\%$, it means that two instances are almost totally overlapped, thus strokes of the below instance are severely shaded. E-Net is still robust in this challenging scenario.

Instance numbers: Similar to overlapping percentile, it is evident that the more instances snarled, the harder the extraction will be. We also select three subsets from our OverlapText-500 dataset, with 50 images for each. for each image, it contains N instances, where N ranges from $[2, 4]$. Experimental results (Row 3 and Row 11∼13 in Table 1) show that MIoU decreases with the increment of overlapped items. As a large N is meaningless and rarely appears in real-world scenes, RecycleNet is sufficient for actual demands.

4.3 Ablation: Silhouette Perfection Network

Exploiting the best configuration of E-Net as a base, we conduct benchmarks to support design of Silhouette Perfection Network (P-Net). As P-Net focuses on recovering the more challenging overlapped pixels, we utilize MIoU on overlapped regions as the major metric. In our testing set, the overlapped pixels occupy about 13% of overall text regions. Notice that the overlapped percentile of instances observably surpass this number. It is the intrinsically hollowed nature of text reduces the percentage.

Expansion operator: The expansion operator is a core design that explicitly encodes strokes knowledge into the network. We first evaluate P-Net with and without the designed operator to show its effectiveness. For P-Net without the designed operator, the expansion convolution is discarded. For fair comparison, we use the original image to replace the expanded label map produced by expansion convolution, in order to keep network volume unchanged. Experimental result (Row 4 and Row 6 in Table 2) demonstrates that without the designed operator, MIoU decreases by 1.05%.

Kernel of expansion operator: The expansion operator has two variables, namely kernel type and kernel size. We first evaluate kernel type selection. Three commonly used kernels, 1) identity kernel, 2) linear decayed kernel and 3) Gaussian kernel, are evaluated. Experimental results (Row 4 and Row 7∼8 in Table 2) demonstrate that Gaussian kernel surpasses others. The Gaussian kernel avoids noise by fetching only certain adjacent candidates.

Size of expansion operator: Basing on the Gaussian kernel, we evaluate the network performance with different kernel sizes. Several regular sizes, 3×3, 5×5, 7×7 are selected for comparison. Experimental results (Row 4 and Row 9∼10 in Table 2) demonstrate that a Gaussian kernel with size 5×5 yields the best balance. This window size allows appropriate expansion to protect vulnerable text strokes.

Loss focuses on overlapped pixels: Loss of P-Net consists of two parts, *i.e.*, the loss of non-overlapped pixels and the loss of overlapped pixels. As P-Net focuses on overlapped pixels recovery, we amplify its weight by λ. We tried different λ and found $\lambda = 2$ is the best balance (Row 4 and Row 11∼12 in Table 2).

Simultaneous training P-Net and E-Net: RecycleNet is designed as an end-to-end trainable network. In the first phase, E-Net and P-Net are trained separately for initialization. In the second phase, two sub-networks are trained jointly. Experimental result (Row 3 and Row 4 in Table 2) demonstrates that a simultaneous training achieves a better performance compared with separated training.

4.4 RecycleNet: Boosting recognition systems

RecycleNet is designed to provide easy and distinct input for following text recognition systems. In this evaluation task, we demonstrate the benefits of RecycleNet when it acts as a front module of text recognition systems. Two mainstream recognition approaches, CTC-based CRNN [29] and attention-based image-to-markup generator [3] are adopted for evaluation. Notice that the selected attention-based approach could directly recognize multi-line texts while CRNN cannot, thus we extract all image crops of single-line instances according to the ground truth, then feed them sequentially to these approaches. We train these two approaches with our synthetic data. An SGD optimizer with learning rate 0.1 is chosen for optimization. The learning rate halves after 300k iterations, and halves again after each 100k iterations. The training batch size is set as 256. For comprehensive evaluation, we exploit outputs in different stages of RecycleNet as input of recognition modules: 1) raw input image (w/o RecycleNet), 2) output of E-Net, 3) output of

Table 2: Ablation study of Silhouette Perfection Network (P-Net).

	Text region					**Overlapped regions in text region**				
Method	Recall	Precision	MAE	MIoU	S-measure	Recall	Precision	MAE	MIoU	S-measure
RecycleNet (E-Net+P-Net)	87.76	89.43	1.35	79.52	94.45	82.21	69.23	0.45	60.21	89.29
P-Net	87.56	88.33	1.43	78.48	93.99	80.53	67.14	0.49	57.76	88.54
E-Net	86.76	88.48	1.46	78.14	93.91	70.88	72.84	0.46	56.07	87.24
w/o expansion operator	86.61	88.90	1.44	78.16	93.92	71.16	73.62	0.45	56.71	87.38
Identity kernel	86.75	89.14	1.42	78.46	94.00	71.52	74.09	0.44	57.21	87.57
Linear decay kernel	86.77	89.34	1.40	78.63	94.08	71.11	74.59	0.44	57.24	87.49
Gaussian kernel (3×3)	86.68	89.02	1.43	78.31	93.97	71.29	74.06	0.44	57.05	87.46
Gaussian kernel (7×7)	86.75	89.08	1.42	78.41	94.01	71.18	74.47	0.44	57.22	87.54
Loss focuses on overlapped pixels ($\lambda = 1$)	87.73	87.95	1.45	78.32	93.93	76.49	69.55	0.47	57.30	88.23
Loss focuses on overlapped pixels ($\lambda = 4$)	89.45	86.20	1.41	78.37	93.82	87.35	60.41	0.51	56.88	88.39

P-Net and 4) output of end-to-end RecycleNet, respectively. Experimental result in Table 3 demonstrates that RecycleNet boosts text recognition performance.

The input format of RecycleNet, *i.e.*, the output format of a front text-detection module is slightly different from the usual provided by mainstream detection systems. Common detection approaches like [5] are designed to precisely extract individual identities. In our scene, such a detection result will probably introduce partial strokes of overlapped instances. Even if RecycleNet could separate them, an extra post-processing cost is required to distinguish and eliminate the meaningless character sequence after recognition. Basing on this observation, RecycleNet requires the text detection module to output two kinds of target: 1) isolated text instances and 2) overlapped text groups that include multiple intersected instances. Compared with the original detection task, we consider our task is no longer more difficult. RecycleNet exploits overlapped groups as input and outputs all separated instances. Existing detection approaches could also be used to provide input for RecycleNet, by simpling merging overlapped detection results as a new target. To confirm this, we conduct benchmarks by using the usual detection format provided by [5] as the input. In this case, RecycleNet also yields competitive result (line 6 in Table 3).

Table 3: Performance of RecycleNet in boosting recognition approaches.

	Recognition method	
Recovery Method	CTC-based [29]	Attention-based [21]
-	78.61	82.42
E-Net	84.04	84.14
P-Net	85.04	85.34
RecycleNet w. usual det.	85.85	85.83
RecycleNet	86.00	86.25

4.5 RecycleNet: Boosting text spotting systems

RecycleNet could also be plugged into text spotting systems to improve their performance. To use RecycleNet, text spotting systems should place it between the detection module and the recognition module, as shown in Fig. 3. However, we observe that recent text spotting systems like [16, 17, 34] are mostly designed as end-to-end trainable, thus there exists no intermediate detection results for RecycleNet to use. Modifying these integrated approaches to insert RecycleNet will introduce uncertainty in evaluation. Facing this problem, we evaluate these systems through a side entry. The evaluation is divided into two steps: 1) Using our testing set as input, RecycleNet processes them and outputs isolated instances. These isolated instances are pasted back orderly and evenly onto a blank background, whose size is the same as the original input. These synthetic images are easy and clear inputs for spotting systems, and they are also counterpart of the original testing set. 2) Original images and synthetic images are sent to spotting systems separately. By replacing original images with synthetic images, the increment of spotting accuracy should be owed to RecycleNet. Experimental result in Table 4 demonstrates that RecycleNet improves recognition accuracy of several recent text spotting systems. Notice that we evaluate selected spotting systems via their released pre-trained models. These models are designed for various purposes and trained with disparate data and volume. Therefore, we are not comparing their performance on our testing set. In this benchmark, we aim to show their relative promotion after adding RecycleNet as a friend.

Table 4: Performance of RecycleNet in boosting text spotting approaches.

Method	w/o RecycleNet	with RecycleNet
ABCNet [17]	17.16	21.68
AE TextSpotter [34]	28.06	42.45
AttentionOCR [44]	28.12	37.91
FOTS [16]	30.60	50.86

5 CONCLUSION

In this paper, we proposed RecycleNet, which automatically extracts and reconstructs all overlapped instances, by fully recycling the intersecting pixels that used to be obstacles for text recognition. RecycleNet parallels to existing text recognition and text spotting systems, and serves as a plug-and-play module for them to achieve better performance. We also released an OverlapText-500 dataset, and wish it could inspire the design of better text recognition and spotting solutions.

REFERENCES

[1] Qifeng Chen and Vladlen Koltun. 2017. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*. 1511–1520.

[2] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. 2018. Pixellink: Detecting scene text via instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[3] Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, and Alexander M Rush. 2017. Image-to-markup generation with coarse-to-fine attention. In *ICML*. JMLR. org, 980–989.

[4] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. 2017. Structure-measure: A New Way to Evaluate Foreground Maps. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 4548–4557. http://dpfan.net/smeasure/.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[6] Pan He, Weilin Huang, Yu Qiao, Chen Loy, and Xiaoou Tang. 2016. Reading scene text in deep convolutional sequences. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[7] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. 2018. An end-to-end textspotter with explicit alignment and attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5020–5029.

[8] Yiqing Hu, Yan Zheng, Hao Liu, Deqiang Jiang, Yinsong Liu, and Bo Ren. 2020. Accurate Structured-Text Spotting for Arithmetical Exercise Correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 686–693.

[9] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Nicolaou, et al. 2015. ICDAR 2015 competition on robust reading. In *ICDAR*. IEEE, 1156–1160.

[10] Chen-Yu Lee and Simon Osindero. 2016. Recursive recurrent nets with attention modeling for ocr in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2231–2239.

[11] Hui Li, Peng Wang, and Chunhua Shen. 2017. Towards end-to-end text spotting with convolutional recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*. 5238–5246.

[12] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. 2020. Mask TextSpotter v3: Segmentation Proposal Network for Robust Scene Text Spotting. *arXiv preprint arXiv:2007.09482* (2020).

[13] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. 2017. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[14] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. 2018. An intriguing failing of convolutional neural networks and the CoordConv solution. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 9628–9639.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.

[16] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. 2018. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5676–5685.

[17] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. 2020. ABCNet: Real-time scene text spotting with adaptive bezier-curve network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9809–9818.

[18] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. 2018. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proceedings of the European conference on computer vision (ECCV)*. 20–36.

[19] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. 2018. Multi-oriented scene text detection via corner localization and region segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7553–7563.

[20] Ke Ma, Zhixin Shu, Xue Bai, Jue Wang, and Dimitris Samaras. 2018. Docunet: document image unwarping via a stacked U-Net. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4700–4709.

[21] Fandong Meng, Zhaopeng Tu, Yong Cheng, Haiyang Wu, Junjie Zhai, Yuekui Yang, and Di Wang. 2018. Neural machine translation with key-value memory-augmented attention. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2574–2580.

[22] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*. Springer, 483–499.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[24] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. 2020. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13528–13537.

[25] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. 2020. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition* 106 (2020), 107404.

[26] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence* 39, 6 (2016), 1137–1149.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[29] Baoguang Shi, Xiang Bai, and Cong Yao. 2016. 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 (2016), 2298–2304.

[30] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. 2016. Detecting text in natural image with connectionist text proposal network. In *European conference on computer vision*. Springer, 56–72.

[31] Fangfang Wang, Yifeng Chen, Fei Wu, and Xi Li. 2020. TextRay: Contour-based Geometric Modeling for Arbitrary-shaped Scene Text Detection. In *Proceedings of the 28th ACM International Conference on Multimedia*. 111–119.

[32] Fangfang Wang, Liming Zhao, Xi Li, Xinchao Wang, and Dacheng Tao. 2018. Geometry-aware scene text detection with instance transformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1381–1389.

[33] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. 2020. Decoupled attention network for text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 12216–12224.

[34] Wenhai Wang, Xuebo Liu, Xiaozhong Ji, Enze Xie, Ding Liang, ZhiBo Yang, Tong Lu, Chunhua Shen, and Ping Luo. 2020. AE TextSpotter: Learning Visual and Linguistic Representation for Ambiguous Text Spotting. In *European Conference on Computer Vision*. Springer, 457–473.

[35] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. 2019. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9336–9345.

[36] Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R Scott. 2019. Convolutional character networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9126–9136.

[37] Xingqian Xu, Zhifei Zhang, Zhaowen Wang, Brian Price, Zhonghao Wang, and Humphrey Shi. 2020. Rethinking Text Segmentation: A Novel Dataset and A Text-Specific Refinement Approach. *arXiv preprint arXiv:2011.14021* (2020).

[38] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1192–1200.

[39] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C Lee Giles. 2017. Learning to read irregular text with attention mechanisms. In *IJCAI*. 3280–3286.

[40] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. 2020. Towards accurate scene text recognition with semantic reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12113–12122.

[41] Fisher Yu and Vladlen Koltun. 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122* (2015).

[42] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. 2020. RobustScanner: Dynamically Enhancing Positional Clues for Robust Text Recognition. In *European Conference on Computer Vision*. Springer, 135–151.

[43] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. 2020. Adaptive Text Recognition through Visual Matching. In *European Conference on Computer Vision*. Springer, 51–67.

[44] Jinjin Zhang, Wei Wang, Di Huang, Qingjie Liu, and Yunhong Wang. 2019. A Feasible Framework for Arbitrary-Shaped Scene Text Recognition. *arXiv preprint arXiv:1912.04561* (2019).

[45] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. 2020. Deep relational reasoning graph network for arbitrary shape text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9699–9708.

[46] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2881–2890.

[47] Yu Zhou, Hongtao Xie, Shancheng Fang, Yan Li, and Yongdong Zhang. 2020. CRNet: A Center-aware Representation for Detecting Text of Arbitrary Shapes. In *Proceedings of the 28th ACM International Conference on Multimedia*. 2571–2580.