International Journal of Computer Vision (2021) 129:161–184
https://doi.org/10.1007/s11263-020-01369-0

Scene Text Detection and Recognition: The Deep Learning Era

Shangbang Long[1] · **Xin He**[2] · **Cong Yao**[3]

Received: 14 April 2020 / Accepted: 8 August 2020 / Published online: 27 August 2020

Abstract
With the rise and development of deep learning, computer vision has been tremendously transformed and reshaped. As an important research area in computer vision, scene text detection and recognition has been inevitably influenced by this wave of revolution, consequentially entering the era of deep learning. In recent years, the community has witnessed substantial advancements in mindset, methodology and performance. This survey is aimed at summarizing and analyzing the major changes and significant progresses of scene text detection and recognition in the deep learning era. Through this article, we devote to: (1) introduce new insights and ideas; (2) highlight recent techniques and benchmarks; (3) look ahead into future trends. Specifically, we will emphasize the dramatic differences brought by deep learning and remaining grand challenges. We expect that this review paper would serve as a reference book for researchers in this field. Related resources are also collected in our Github repository (https://github.com/Jyouhou/SceneTextPapers).

Keywords Scene text · Optical character recognition · Detection · Recognition · Deep learning · Survey

1 Introduction

Undoubtedly, text is among the most brilliant and influential creations of humankind. As the written form of human languages, text makes it feasible to reliably and effectively spread or acquire information across time and space. In this sense, text constitutes the cornerstone of human civilization.

On the one hand, as a vital tool for communication and collaboration, text has been playing a more important role than ever in modern society; on the other hand, the rich and precise high-level semantics embodied in text could be beneficial for understanding the world around us. For example, text information can be used in a wide range of real-world applications, such as *image search* (Tsai et al. 2011; Schroth et al. 2011), *instant translation* (Dvorin and Havosha 2009; Parkinson et al. 2016), *robots navigation* (DeSouza and Kak 2002; Liu and Samarabandu 2005a, b; Schulz et al. 2015), and *industrial automation* (Ham et al. 1995; He et al. 2005; Chowdhury and Deb 2013). Therefore, automatic text reading from natural environments, as illustrated in Fig. 1, a.k.a. scene text detection and recognition (Zhu et al. 2016) or PhotoOCR (Bissacco et al. 2013), has become an increasing popular and important research topic in computer vision.

However, despite years of research, a series of grand challenges may still be encountered when detecting and recognizing text in the wild. The difficulties mainly stem from three aspects:

- *Diversity and Variability of Text in Natural Scenes* Distinctive from scripts in documents, text in natural scene exhibit much higher diversity and variability. For example, instances of scene text can be in different languages, colors, fonts, sizes, orientations, and shapes. Moreover, the aspect ratios and layouts of scene text may vary significantly. All these variations pose challenges for detection and recognition algorithms designed for text in natural scenes.
- *Complexity and Interference of Backgrounds* The backgrounds of natural scenes are virtually unpredictable. There might be patterns extremely similar to text (e.g.,

Communicated by Vittorio Ferrari.

✉ Shangbang Long
shangbal@cs.cmu.edu

Xin He
hexin7257@gmail.com

Cong Yao
yaocong2010@gmail.com

[1] Machine Learning Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA

[2] ByteDance Ltd, Beijing, China

[3] MEGVII Inc. (Face++), Beijing, China

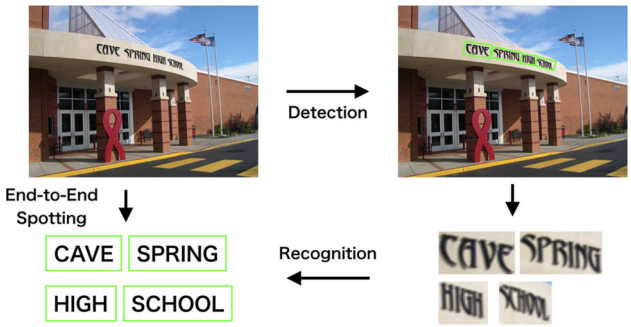

Fig. 1 Schematic diagram of scene text detection and recognition. The image sample is from total-text (Ch'ng and Chan 2017).

tree leaves, traffic signs, bricks, windows, and stockades), or occlusions caused by foreign objects, which may potentially lead to confusion and mistakes.
- *Imperfect Imaging Conditions* In uncontrolled circumstances, the quality of text images and videos could not be guaranteed. That is, in poor imaging conditions, text instances may be of low resolution and severe distortion due to inappropriate shooting distance or angle, or blurred because of out of focus or shaking, or noised on account of low light level, or corrupted by highlights or shadows.

These difficulties run through the years before deep learning showed its potential in computer vision as well as in other fields. As deep learning came to prominence after AlexNet (Krizhevsky et al. 2012) won the ILSVRC2012 (Russakovsky et al. 2015) contest, researchers turn to deep neural networks for automatic feature learning and start with more in-depth studies. The community are now working on ever more challenging targets. The progress made in recent years can be summarized as follows:

- *Incorporation of Deep Learning* Nearly all recent methods are built upon deep learning models. Most importantly, deep learning frees researchers from the exhausting work of repeatedly designing and testing hand-crafted features, which gives rise to a blossom of works that push the envelope further. To be specific, the use of deep learning substantially simplifies the overall pipeline, as illustrated in Fig. 3. Besides, these algorithms provide significant improvements over previous ones on standard benchmarks. Gradient-based training routines also facilitate to end-to-end trainable methods.
- *Challenge-Oriented Algorithms and Datasets* Researchers are now turning to more specific aspects and challenges. Against difficulties in real-world scenarios, newly published datasets are collected with unique and representative characteristics. For example, there are datasets featuring long text (Tu et al. 2012), blurred text (Karatzas et al. 2015), and curved text (Ch'ng and Chan 2017) respectively. Driven by these datasets, almost all algorithms published in recent years are designed to tackle specific challenges. For instance, some are proposed to detect oriented text, while others aim at blurred and unfocused scene images. These ideas are also combined to make more general-purpose methods.
- *Advances in Auxiliary Technologies* Apart from new datasets and models devoted to the main task, auxiliary technologies that do not solve the task directly also find their places in this field, such as synthetic data and bootstrapping.

In this survey, we present an overview of the recent development in deep-learning-based text detection and recognition from still scene images. We review methods from different perspectives and list the up-to-date datasets. We also analyze the status quo and future research trends.

There have been already several excellent review papers (Uchida 2014; Ye and Doermann 2015; Yin et al. 2016; Zhu et al. 2016), which also organize and analyze works related to text detection and recognition. However, these papers are published before deep learning came to prominence in this field. Therefore, they mainly focus on more traditional and feature-based methods. We refer readers to these paper as well for a more comprehensive view and knowledge of the history of this field. This article will mainly concentrate on text information extraction from still images, rather than videos. For scene text detection and recognition in videos, please also refer to Jung et al. (2004) and Yin et al. (2016).

The remaining parts of this paper are arranged as follows: In Sect. 2, we briefly review the methods before the deep learning era. In Sect. 3, we list and summarize algorithms based on deep learning in a hierarchical order. Note that we do not introduce these techniques in a paper-by-paper order, but instead based on a taxonomy of their methodologies. Some papers may appear in several sections if they have contributions to multiple aspects. In Sect. 4, we take a look at the datasets and evaluation protocols. Finally, in Sects. 5 and 6, we present potential applications and our own opinions on the current status and future trends.

2 Methods Before the Deep Learning Era

In this section, we take a glance retrospectively at algorithms before the deep learning era. More detailed and comprehensive coverage of these works can be found in Uchida (2014), Ye and Doermann (2015), Yin et al. (2016), and Zhu et al. (2016). For text detection and recognition, the attention has been the design of features.

In this period of time, most text detection methods either adopt Connected Components Analysis (CCA) (Huang et al.

Fig. 2 Illustration of traditional methods with hand-crafted features: (1) Maximally Stable Extremal Regions (MSER) (Neumann and Matas 2010), assuming chromatic consistency within each character; (2) Stroke Width Transform (SWT) (Epshtein et al. 2010), assuming consistent stroke width within each character

2013; Neumann and Matas 2010; Epshtein et al. 2010; Tu et al. 2012; Yin et al. 2014; Yi and Tian 2011; Jain and Yu 1998) or Sliding Window (SW) based classification (Lee et al. 2011; Wang et al. 2011; Coates et al. 2011; Wang et al. 2012). CCA based methods first extract candidate components through a variety of ways (e.g., color clustering or extreme region extraction), and then filter out non-text components using manually designed rules or classifiers automatically trained on hand-crafted features (see Fig. 2). In sliding window classification methods, windows of varying sizes slide over the input image, where each window is classified as text segments/regions or not. Those classified as positive are further grouped into text regions with morphological operations (Lee et al. 2011), Conditional Random Field (CRF) (Wang et al. 2011) and other alternative graph based methods (Coates et al. 2011; Wang et al. 2012).

For text recognition, one branch adopted the feature-based methods. Shi et al. (2013) and Yao et al. (2014) propose character segments based recognition algorithms. Rodriguez-Serrano et al. (2013), Rodriguez-Serrano et al. (2015), Gordo (2015), Almazán et al. (2014) utilize label embedding to directly perform matching between strings and images. Strokes (Busta et al. 2015) and character key-points (Quy Phan et al. 2013) are also detected as features for classification. Another decomposes the recognition process into a series of sub-problems. Various methods have been proposed to tackle these sub-problems, which includes text binarization (Zhiwei et al. 2010; Mishra et al. 2011; Wakahara and Kita 2011; Lee and Kim 2013), text line segmentation (Ye et al. 2003), character segmentation (Nomura et al. 2005; Shivakumara et al. 2011; Roy et al. 2009), single character recognition (Chen et al. 2004; Sheshadri and Divvala 2012) and word correction (Zhang and Chang 2003; Wachenfeld et al. 2006; Mishra et al. 2012; Karatzas and Antonacopoulos 2004; Weinman et al. 2007).

There have been efforts devoted to integrated (i.e. end-to-end as we call it today) systems as well (Wang et al. 2011; Neumann and Matas 2013). In Wang et al. (2011), characters are considered as a special case in object detection and detected by a nearest-neighbor classifier trained on HOG features (Dalal and Triggs 2005) and then grouped into words through a Pictorial Structure (PS) based model (Felzenszwalb and Huttenlocher 2005). Neumann and Matas (Neumann and Matas 2013) proposed a decision delay approach by keeping multiple segmentations of each character until the last stage when the context of each character is known. They detect character segmentation using extremal regions and decode recognition results through a dynamic programming algorithm.

In summary, text detection and recognition methods before the deep learning era mainly extract low-level or mid-level handcrafted image features, which entails demanding and repetitive pre-processing and post-processing steps. Constrained by the limited representation ability of handcrafted features and the complexity of pipelines, those methods can hardly handle intricate circumstances, e.g. blurred images in the ICDAR 2015 dataset (Karatzas et al. 2015).

3 Methodology in the Deep Learning Era

As implied by the title of this section, we would like to address recent advances as changes in *methodology* instead of merely new *methods*. Our conclusion is grounded in the observations as explained in the following paragraph.

Methods in the recent years are characterized by the following two distinctions: (1) Most methods utilize deep-learning based models; (2) Most researchers are approaching the problem from a diversity of perspectives, trying to solve different challenges. Methods driven by deep learning enjoy the advantage that automatic feature learning can save us from designing and testing a large amount of potential handcrafted features. At the same time, researchers from different viewpoints are enriching and promoting the community into more in-depth work, aiming at different targets, e.g. faster and simpler pipeline (Zhou et al. 2017), text of varying aspect ratios (Shi et al. 2017a), and synthetic data (Gupta et al. 2016). As we can also see further in this section, the incorporation of deep learning has totally changed the way researchers approach the task and has enlarged the scope of

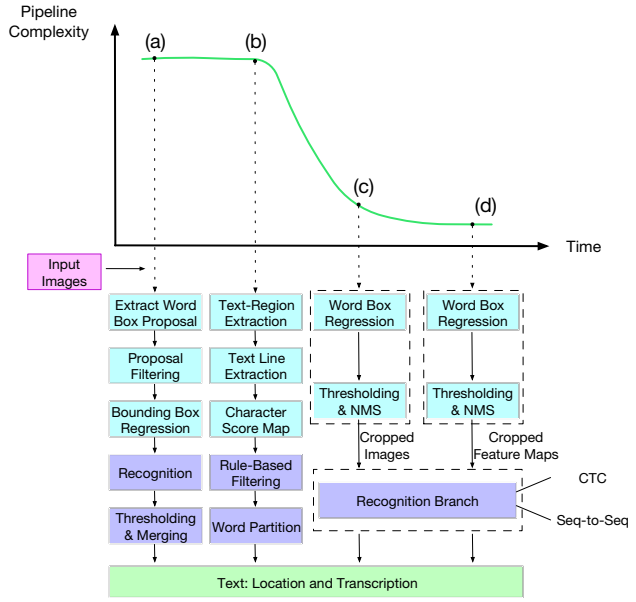

Fig. 3 Illustrations of representative scene text detection and recognition system pipelines. **a** Jaderberg et al. (2016) and **b** Yao et al. (2016) are representative multi-step methods. **c**, **d** are simplified pipeline. In **c**, detectors and recognizers are separate. In **d**, the detectors pass cropped feature maps to recognizers, which allows end-to-end training

research by far. This is the most significant change compared to the former epoch.

In this section, we would classify existing methods into a hierarchical taxonomy, and introduce them in a top-down style. First, we divide them into four kinds of systems: (1) text detection that detects and localizes text in natural images; (2) recognition system that transcribes and converts the content of the detected text regions into linguistic symbols; (3) end-to-end system that performs both text detection and recognition in one unified pipeline; (4) auxiliary methods that aim to support the main task of text detection and recognition, e.g. synthetic data generation. Under each category, we review recent methods from different perspectives.

3.1 Detection

We acknowledge that scene text detection can be taxonomically subsumed under general object detection, which is dichotomized as one-staged methods and two-staged ones. Indeed, many scene text detection algorithms are majorly inspired by and follow the designs of general object detectors. Therefore we also encourage readers to refer to recent surveys on object detection methods (Han et al. 2018; Liu et al. 2018a). However, the detection of scene text has a different set of characteristics and challenges that require unique methodologies and solutions. Thus, many methods rely on special representation for scene text to solve these non-trivial problems.

The evolution of scene text detection algorithms, therefore, undergoes three main stages: (1) In the first stage, learning-based methods are equipped with multi-step pipelines, but these methods are still slow and complicated. (2) Then, the idea and methods of general object detection are successfully implanted into this task. (3) In the third stage, researchers design special representations based on sub-text components to solve the challenges of long text and irregular text.

3.1.1 Early Attempts to Utilize Deep Learning

Early deep-learning-based methods (Huang et al. 2014; Tian et al. 2015; Yao et al. 2016; Zhang et al. 2016; He et al. 2017a) approach the task of text detection into a multi-step process. They use convolutional neural networks (CNNs) to predict local segments and then apply heuristic post-processing steps to merge segments into detection lines.

In an early attempt (Huang et al. 2014), CNNs are only used to classify local image patches into text and non-text classes. They propose to mine such image patches using MSER features. Positive patches are then merged into text lines.

Later, CNNs are applied to the whole images in a fully convolutional approach. TextFlow (Tian et al. 2015) uses CNNs to detect character and views the character grouping task as a min-cost flow problem (Goldberg 1997).

In Yao et al. (2016), a convolutional neural network is used to predict whether each pixel in the input image (1) belongs to characters, (2) is inside the text region, and (3) the text orientation around the pixel. Connected positive responses are considered as detected characters or text regions. For characters belonging to the same text region, Delaunay triangulation (Kang et al. 2014) is applied, after which a graph partition algorithm groups characters into text lines based on the predicted orientation attribute.

Similarly, Zhang et al. (2016) first predicts a segmentation map indicating text line regions. For each text line region, MSER (Neumann and Matas 2012) is applied to extract character candidates. Character candidates reveal information on the scale and orientation of the underlying text line. Finally, minimum bounding boxes are extracted as the final text line candidates.

He et al. (2017a) propose a detection process that also consists of several steps. First, text blocks are extracted. Then the model crops and only focuses on the extracted text blocks to extract text center line (TCL), which is defined as a shrunk version of the original text line. Each text line represents the existence of one text instance. The extracted TCL map is then split into several TCLs. Each split TCL is then concatenated to the original image. A semantic segmentation model then classifies each pixel into ones that belong to the same text instance as the given TCL, and ones that do not.

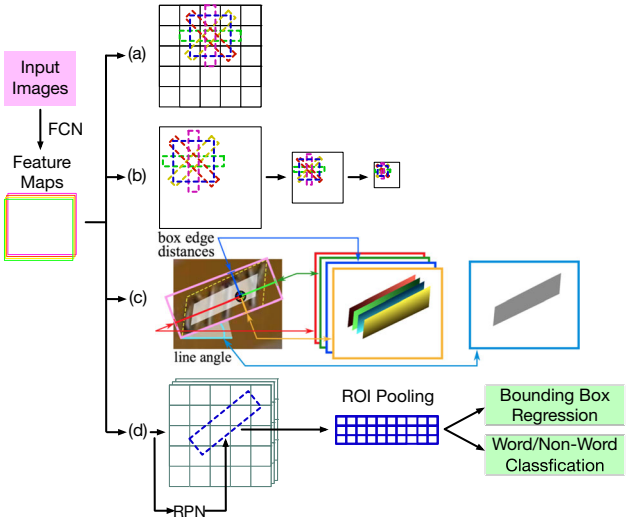

Fig. 4 High-level illustration of methods inspired by general object detection: **a** Similar to YOLO (Redmon et al. 2016), regressing offsets based on default bounding boxes at each anchor position. **b** Variants of SSD (Liu et al. 2016a), predicting at feature maps of different scales. **c** Predicting at each anchor position and regressing the bounding box directly. **d** Two-staged methods with an extra stage to correct the initial regression results

Overall, in this stage, scene text detection algorithms still have long and slow pipelines, though they have replaced some hand-crafted features with learning-based ones. The design methodology is bottom-up and based on key components, such as single characters and text center lines.

3.1.2 Methods Inspired by Object Detection

Later, researchers are drawing inspirations from the rapidly developing general object detection algorithms (Liu et al. 2016a; Fu et al. 2017; Girshick et al. 2014; Girshick 2015; Ren et al. 2015; He et al. 2017b). In this stage, scene text detection algorithms are designed by modifying the region proposal and bounding box regression modules of general detectors to localize text instances directly (Dai et al. 2017; He et al. 2017c; Jiang et al. 2017; Liao et al. 2017, 2018a; Liu and Jin 2017; Shi et al. 2017a; Liu et al. 2017; Ma et al. 2017; Li et al. 2017b; Liao et al. 2018b; Zhang et al. 2018), as shown in Fig. 4. They mainly consist of stacked convolutional layers that encode the input images into feature maps. Each spatial location at the feature map corresponds to a region of the input image. The feature maps are then fed into a classifier to predict the existence and localization of text instances at each such spatial location.

These methods greatly reduce the pipeline into an end-to-end trainable neural network component, making training much easier and inference much faster. We introduce the most representative works here.

Inspired by one-staged object detectors, TextBoxes (Liao et al. 2017) adapts SSD (Liu et al. 2016a) to fit the varying orientations and aspect-ratios of text by defining default boxes as quadrilaterals with different aspect-ratio specs.

EAST (Zhou et al. 2017) further simplifies the anchor-based detection by adopting the U-shaped design (Ronneberger et al. 2015) to integrate features from different levels. Input images are encoded as one multi-channeled feature map instead of multiple layers of different spatial sizes in SSD. The feature at each spatial location is used to regress the rectangular or quadrilateral bounding box of the underlying text instances directly. Specifically, the existence of text, i.e. text/non-text, and geometries, e.g. orientation and size for rectangles, and vertexes coordinates for quadrilaterals, are predicted. EAST makes a difference to the field of text detection with its highly simplified pipeline and efficiency to perform inference at real-time speed.

Other methods adapt the two-staged object detection framework of R-CNN (Girshick et al. 2014; Girshick 2015; Ren et al. 2015), where the second stage corrects the localization results based on features obtained by Region of Interest (RoI) pooling.

In Ma et al. (2017), rotation region proposal networks are adapted to generate rotating region proposals, in order to fit into text of arbitrary orientations, instead of axis-aligned rectangles.

In FEN (Zhang et al. 2018), the weighted sum of RoI poolings with different sizes is used. The final prediction is made by leveraging the *textness* score for poolings of 4 different sizes.

Zhang et al. (2019) propose to perform RoI and localization branch recursively, to revise the predicted position of the text instance. It is a good way to include features at the boundaries of bounding boxes, which localizes the text better than region proposal networks (RPNs).

Wang et al. (2018) propose to use a parametrized *Instance Transformation Network* (ITN) that learns to predict appropriate affine transformation to perform on the last feature layer extracted by the base network, to rectify oriented text instances. Their method, with ITN, can be trained end-to-end.

To adapt to irregularly shaped text, bounding polygons (Liu et al. 2017) with as many as 14 vertexes are proposed, followed by a Bi-LSTM (Hochreiter and Schmidhuber 1997) layer to refine the coordinates of the predicted vertexes.

In a similar way, Wang et al. (2019b) propose to use recurrent neural networks (RNNs) to read the features encoded by RPN-based two-staged object decoders and predict the bounding polygon with variable length. The method requires no post-processing or complex intermediate steps and achieves a much faster speed of 10.0 FPS on Total-Text.

The main contribution in this stage is the simplification of the detection pipeline and the following improvement of efficiency. However, the performance is still limited when faced

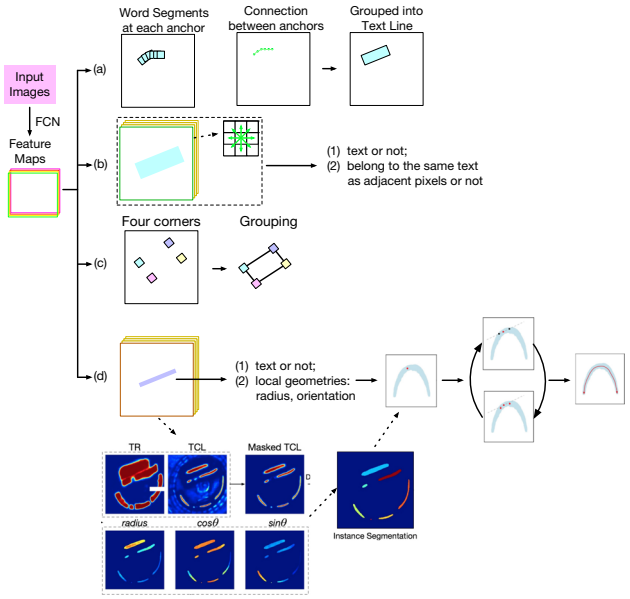

Fig. 5 Illustration of representative methods based on sub-text components: **a** SegLink (Shi et al. 2017a): with SSD as base network, predict word segments at each anchor position, and connections between adjacent anchors. **b** PixelLink (Deng et al. 2018): for each pixel, predict text/non-text classification and whether it belongs to the same text as adjacent pixels or not. **c** Corner Localization (Lyu et al. 2018b): predict the four corners of each text and group those belonging to the same text instances. **d** TextSnake (Long et al. 2018): predict text/non-text and local geometries, which are used to reconstruct text instance

with curved, oriented, or long text for one-staged methods due to the limitation of the receptive field, and the efficiency is limited for two-staged methods.

3.1.3 Methods Based on Sub-text Components

The main distinction between text detection and general object detection is that text is homogeneous as a whole and is characterized by its locality, which is different from general object detection. By homogeneity and locality, we refer to the property that any part of a text instance is still text. Humans do not have to see the whole text instance to know it belongs to some text.

Such a property lays a cornerstone for a new branch of text detection methods that only predict sub-text components and then assemble them into a text instance. These methods, by its nature, can better adapt to the aforementioned challenges of curved, long, and oriented text. These methods, as illustrated in Fig. 5, use neural networks to predict local attributes or segments, and a post-processing step to re-construct text instances. Compared with early multi-staged methods, they rely more on neural networks and have shorter pipelines.

In **pixel-level** methods (Deng et al. 2018; Wu and Natarajan 2017), an end-to-end fully convolutional neural network learns to generate a dense prediction map indicating whether each pixel in the original image belongs to any text instances or not. Post-processing methods then group pixels together depending on which pixels belong to the same text instance. Basically, they can be seen as a special case of instance segmentation (He et al. 2017b). Since text can appear in clusters which makes predicted pixels connected to each other, the core of pixel-level methods is to separate text instances from each other.

PixelLink (Deng et al. 2018) learns to predict whether two adjacent pixels belong to the same text instance by adding extra output channels to indicate links between adjacent pixels.

Border learning method (Wu and Natarajan 2017) casts each pixel into three categories: text, border, and background, assuming that the border can well separate text instances.

In Wang et al. (2017), pixels are clustered according to their color consistency and edge information. The fused image segments are called *superpixel*. These superpixels are further used to extract characters and predict text instances.

Upon the segmentation framework, Tian et al. (2019) propose to add a loss term that maximizes the Euclidean distances between pixel embedding vectors that belong to different text instances, and minimizes those belonging to the same instance, to better separate adjacent texts.

Wang et al. (2019a) propose to predict text regions at different shrinkage scales, and enlarges the detected text region round-by-round, until collision with other instances. However, the prediction at different scales is itself a variation of the aforementioned border learning (Wu and Natarajan 2017).

Component-level methods usually predict at a medium granularity. Component refers to a local region of text instance, sometimes overlapping one or more characters.

The representative component-level method is Connectionist Text Proposal Network (CTPN) (Tian et al. 2016). CTPN models inherit the idea of anchoring and recurrent neural network for sequence labeling. They stack an RNN on top of CNNs. Each position in the final feature map represents features in the region specified by the corresponding anchor. Assuming that text appears horizontally, each row of features are fed into an RNN and labeled as text/non-text. Geometries such as segment sizes are also predicted. CTPN is the first to predict and connect segments of scene text with deep neural networks.

SegLink (Shi et al. 2017a) extends CTPN by considering the multi-oriented linkage between segments. The detection of segments is based on SSD (Liu et al. 2016a), where each default box represents a *text segment*. Links between default boxes are predicted to indicate whether the adjacent segments belong to the same text instance. Zhang et al. (2020) further improve SegLink by using a Graph Convolutional Network (Kipf and Welling 2016) to predict the linkage between segments.

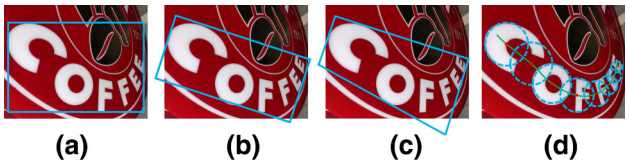

Fig. 6 **a**–**c** Representing text as horizontal rectangles, oriented rectangles, and quadrilaterals. **d** The sliding-disk representation proposed in TextSnake (Long et al. 2018)

Corner localization method (Lyu et al. 2018b) proposes to detect the four corners of each text instance. Since each text instance only has 4 corners, the prediction results and their relative position can indicate which corners should be grouped into the same text instance.

Long et al. (2018) argue that text can be represented as a series of sliding round disks along the text center line (TCL), which is in accord with the running direction of the text instance, as shown in Fig. 6. With the novel representation, they present a new model, *TextSnake*, which learns to predict local attributes, including TCL/non-TCL, text-region/non-text-region, radius, and orientation. The intersection of TCL pixels and text region pixels gives the final prediction of pixel-level TCL. Local geometries are then used to extract the TCL in the form of an ordered point list. With TCL and radius, the text line is reconstructed. It achieves state-of-the-art performance on several curved text datasets as well as more widely used ones, e.g. ICDAR 2015 (Karatzas et al. 2015) and MSRA-TD 500 (Tu et al. 2012). Notably, Long et al. propose a cross-validation test across different datasets, where models are only fine-tuned on datasets with straight text instances and tested on the curved datasets. In all existing curved datasets, TextSnake achieves improvements by up to 20% over other baselines in F1-Score.

Character-level representation is yet another effective way. Baek et al. (2019b) propose to learn a segmentation map for character centers and links between them. Both components and links are predicted in the form of a Gaussian heat map. However, this method requires iterative weak supervision as real-world datasets are rarely equipped with character-level labels.

Overall, detection based on sub-text components enjoys better flexibility and generalization ability over shapes and aspect ratios of text instance. The main drawback is that the module or post-processing step used to group segments into text instances may be vulnerable to noise, and the efficiency of this step is highly dependent on the actual implementation, and therefore may vary among different platforms.

3.2 Recognition

In this section, we introduce methods for *scene text recognition*. The input of these methods is cropped text instance images which contain only one word.

In the deep learning era, scene text recognition models use CNNs to encode images into feature spaces. The main difference lies in the text content decoding module. Two major techniques are the Connectionist Temporal Classification (Graves et al. 2006) (CTC) and the encoder–decoder framework (Sutskever et al. 2014). We introduce recognition methods in the literature based on the main technique they employ. Mainstream frameworks are illustrated in Fig. 7.

Both CTC and encoder–decoder frameworks are originally designed for 1-dimensional sequential input data, and therefore are applicable to the recognition of straight and horizontal text, which can be encoded into a sequence of feature frames by CNNs without losing important information. However, characters in oriented and curved text are distributed over a 2-dimensional space. It remains a challenge to effectively represent oriented and curved text in feature spaces in order to fit the CTC and encoder–decoder frameworks, whose decodes require 1-dimensional inputs. For oriented and curved text, directly compressing the features into a 1-dimensional form may lose relevant information and bring in noise from background, thus leading to inferior recognition accuracy. We would introduce techniques to solve this challenge.

3.2.1 CTC-Based Methods

The CTC decoding module is adopted from speech recognition, where data are sequential in the time domain. To apply CTC in scene text recognition, the input images are viewed as a sequence of vertical pixel frames. The network outputs a per-frame prediction, indicating the probability distribution of label types for each frame. The CTC rule is then applied to edit the per-frame prediction to a text string. During training, the loss is computed as the sum of the negative log probability of all possible per-frame predictions that can generate the target sequence by CTC rules. Therefore, the CTC method makes it end-to-end trainable with only word-level annotations, without the need for character level annotations. The first application of CTC in the OCR domain can be traced to the handwriting recognition system of Graves et al. (2008). Now this technique is widely adopted in scene text recognition (Su and Lu 2014; He et al. 2016; Liu et al. 2016b; Gao et al. 2017; Shi et al. 2017b; Yin et al. 2017).

The first attempts can be referred to as convolutional recurrent neural networks (CRNN). These models are composed by stacking RNNs on top of CNNs and use CTC for training and inference. DTRN (He et al. 2016) is the first CRNN model. It slides a CNN model across the input images to generate convolutional feature slices, which are then fed into RNNs. Shi et al. (2017b) further improves DTRN by adopting the fully convolutional approach to encode the input images as a whole to generate features slices, utilizing the property that CNNs are not restricted by the spatial sizes of inputs.

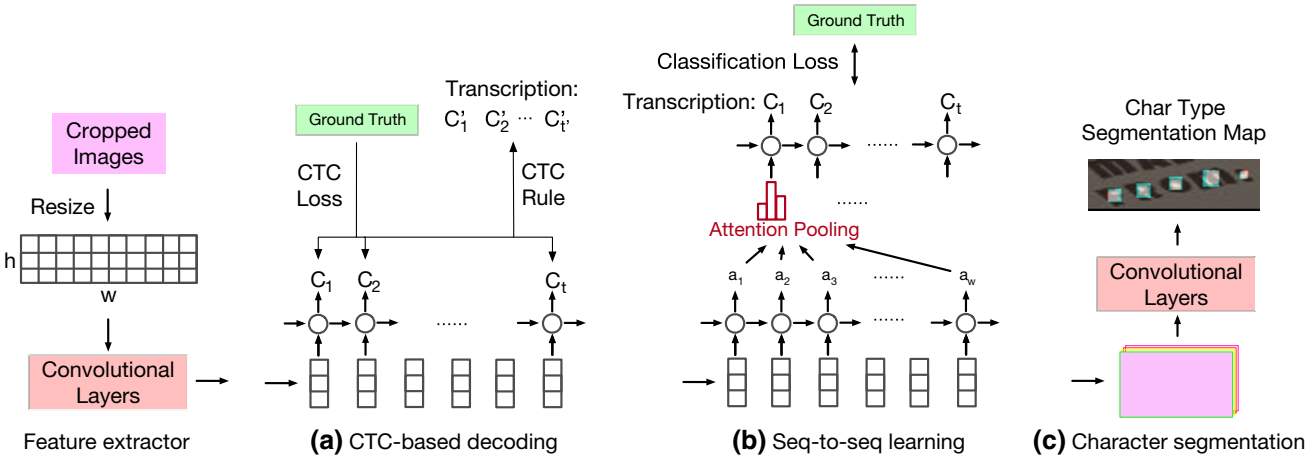

Fig. 7 Frameworks of text recognition models. **a** Sequence tagging model, and uses CTC for alignment in training and inference. **b** Sequence to sequence model, and can use cross-entropy to learn directly. **c** Segmentation-based methods

Instead of RNN, Gao et al. (2017) adopt the stacked convolutional layers to effectively capture the contextual dependencies of the input sequence, which is characterized by lower computational complexity and easier parallel computation.

Yin et al. (2017) simultaneously detect and recognize characters by sliding the text line image with character models, which are learned end-to-end on text line images labeled with text transcripts.

3.2.2 Encoder–Decoder Methods

The encoder–decoder framework for sequence-to-sequence learning is originally proposed in Sutskever et al. (2014) for machine translation. The encoder RNN reads an input sequence and passes its final latent state to a decoder RNN, which generates output in an auto-regressive way. The main advantage of the encoder–decoder framework is that it gives outputs of variable lengths, which satisfies the task setting of scene text recognition. The encoder–decoder framework is usually combined with the attention mechanism (Bahdanau et al. 2014) which jointly learns to align input sequence and output sequence.

Lee and Osindero (2016) present recursive recurrent neural networks with attention modeling for lexicon-free scene text recognition. the model first passes input images through recursive convolutional layers to extract encoded image features and then decodes them to output characters by recurrent neural networks with implicitly learned character-level language statistics. The attention-based mechanism performs soft feature selection for better image feature usage.

Cheng et al. (2017a) observe the attention drift problem in existing attention-based methods and proposes to impose localization supervision for attention score to attenuate it.

Bai et al. (2018) propose an edit probability (EP) metric to handle the misalignment between the ground truth string and the attention's output sequence of the probability distribution. Unlike aforementioned attention-based methods, which usually employ a frame-wise maximal likelihood loss, EP tries to estimate the probability of generating a string from the output sequence of probability distribution conditioned on the input image, while considering the possible occurrences of missing or superfluous characters.

Liu et al. (2018d) propose an efficient attention-based encoder–decoder model, in which the encoder part is trained under binary constraints to reduce computation cost.

Both CTC and the encoder–decoder framework simplify the recognition pipeline and make it possible to train scene text recognizers with only word-level annotations instead of character level annotations. Compared to CTC, the decoder module of the encoder–decoder framework is an implicit language model, and therefore, it can incorporate more linguistic priors. For the same reason, the encoder–decoder framework requires a larger training dataset with a larger vocabulary. Otherwise, the model may degenerate when reading words that are unseen during training. On the contrary, CTC is less dependent on language models and has a better character-to-pixel alignment. Therefore it is potentially better on languages such as Chinese and Japanese that have a large character set. The main drawback of these two methods is that they assume the text to be straight, and therefore can not adapt to irregular text.

3.2.3 Adaptions for Irregular Text Recognition

Rectification-modules are a popular solution to irregular text recognition. Shi et al. (2016, 2018) propose a text recognition system which combined a Spatial Transformer Network (STN) (Jaderberg et al. 2015) and an attention-based

Sequence Recognition Network. The STN-module predicts text bounding polygons with fully connected layers in order for Thin-Plate-Spline transformations which rectify the input irregular text image into a more canonical form, i.e. straight text. The rectification proves to be a successful strategy and forms the basis of the winning solution (Long et al. 2019) in ICDAR 2019 ArT[1] irregular text recognition competition.

There have also been several improved versions of rectification based recognition. Zhan and Lu (2019) propose to perform rectification multiple times to gradually rectify the text. They also replace the text bounding polygons with a polynomial function to represent the shape. Yang et al. (2019) propose to predict local attributes, such as radius and orientation values for pixels inside the text center region, in a similar way to TextSnake (Long et al. 2018). The orientation is defined as the orientation of the underlying character boxes, instead of text bounding polygons. Based on the attributes, bounding polygons are reconstructed in a way that the perspective distortion of characters is rectified, while the method by Shi et al. and Zhan et al. may only rectify at the text level and leave the characters distorted.

Yang et al. (2017) introduce an auxiliary dense character detection task to encourage the learning of visual representations that are favorable to the text patterns. And they adopt an alignment loss to regularize the estimated attention at each time-step. Further, they use a coordinate map as a second input to enforce spatial-awareness.

Cheng et al. (2017b) argue that encoding a text image as a 1-D sequence of features as implemented in most methods is not sufficient. They encode an input image to four feature sequences of four directions: horizontal, reversed horizontal, vertical, and reversed vertical. A weighting mechanism is applied to combine the four feature sequences.

Liu et al. (2018b) present a hierarchical attention mechanism (HAM) which consists of a recurrent RoI-Warp layer and a character-level attention layer. They adopt a local transformation to model the distortion of individual characters, resulting in improved efficiency, and can handle different types of distortion that are hard to be modeled by a single global transformation.

Liao et al. (2019b) cast the task of recognition into semantic segmentation, and treat each character type as one class. The method is insensitive to shapes and is thus effective on irregular text, but the lack of end-to-end training and sequence learning makes it prone to single-character errors, especially when the image quality is low. They are also the first to evaluate the robustness of their recognition method by padding and transforming test images.

Another solution to irregular scene text recognition is 2-dimensional attention (Xu et al. 2015), which has been verified in Li et al. (2019). Different from the sequential encoder–decoder framework, the 2D attentional model maintains 2-dimensional encoded features, and attention scores are computed for all spatial locations. Similar to spatial attention, Long et al. (2020) propose to first detect characters. Afterward, features are interpolated and gathered along the character center lines to form sequential feature frames.

In addition to the aforementioned techniques, Qin et al. (2019) show that simply flattening the feature maps from 2-dimensional to 1-dimensional and feeding the resulting sequential features to RNN based attentional encoder–decoder model is sufficient to produce state-of-the-art recognition results on irregular text, which is a simple yet efficient solution.

Apart from tailored model designs, Long et al. (2019) synthesizes a curved text dataset, which significantly boosts the recognition performance on real-world curved text datasets with no sacrifices to straight text datasets.

Although many elegant and neat solutions have been proposed, they are only evaluated and compared based on a relatively small dataset, CUTE80, which only consists of 288 word samples. Besides, the training datasets used in these works only contain a negligible proportion of irregular text samples. Evaluations on larger datasets and more suitable training datasets may help us understand these methods better.

3.2.4 Other Methods

Jaderberg et al. (2014a, b) perform word recognition by classifying the image into a pre-defined set of vocabulary, under the framework of image classification. The model is trained by synthetic images, and achieves state-of-the-art performance on some benchmarks containing English words only. However, the application of this method is quite limited as it cannot be applied to recognize unseen sequences such as phone numbers and email addresses.

To improve performance on difficult cases such as occlusion which brings ambiguity to single character recognition, Yu et al. (2020) propose a transformer-based semantic reasoning module that performs translations from coarse, prone-to-error text outputs from the decoder to fine and linguistically calibrated outputs, which bears some resemblance to the deliberation networks for machine translation (Xia et al. 2017) that first translate and then re-write the sentences.

Despite the progress we have seen so far, the evaluation of recognition methods falls behind the time. As most detection methods can detect oriented and irregular text and some even rectify them, the recognition of such text may seem redundant. On the other hand, the robustness of recognition when cropped with a slightly different bounding box is seldom verified. Such robustness may be more important in real-world scenarios.

[1] https://rrc.cvc.uab.es/?ch=14.

3.3 End-to-End System

In the past, text detection and recognition are usually cast as two independent sub-problems that are combined to perform text reading from images. Recently, many end-to-end text detection and recognition systems (also known as text spotting systems) have been proposed, profiting a lot from the idea of designing differentiable computation graphs, as shown in Fig. 8. Efforts to build such systems have gained considerable momentum as a new trend.

Two-Step Pipelines While earlier work (Wang et al. 2011, 2012) first detect single characters in the input image, recent systems usually detect and recognize text in word-level or line level. Some of these systems first generate text proposals using a text detection model and then recognize them with another text recognition model (Jaderberg et al. 2016; Liao et al. 2017; Gupta et al. 2016). Jaderberg et al. (2016) use a combination of Edge Box proposals (Zitnick and Dollár 2014) and a trained aggregate channel features detector (Dollár et al. 2014) to generate candidate word bounding boxes. Proposal boxes are filtered and rectified before being sent into their recognition model proposed in (Jaderberg et al. 2014b). Liao et al. (2017) combine an SSD (Liu et al. 2016a) based text detector and CRNN (Shi et al. 2017b) to spot text in images.

In these methods, detected words are cropped from the image, and therefore, the detection and recognition are two separate steps. One major drawback of the two-step methods is that the propagation of error between the detection and recognition models will lead to less satisfactory performance.

Two-Stage Pipelines Recently, end-to-end trainable networks are proposed to tackle this problem (Bartz et al. 2017; Busta et al. 2017; Li et al. 2017a; He et al. 2018; Liu et al. 2018c), where feature maps instead of images are cropped and fed to recognition modules.

Bartz et al. (2017) present an solution that utilizes a STN (Jaderberg et al. 2015) to circularly attend to each word in the input image, and then recognize them separately. The united network is trained in a weakly-supervised manner that no word bounding box labels are used. Li et al. (2017a) substitute the object classification module in Faster-RCNN (Ren et al. 2015) with an encoder–decoder based text recognition model and make up their text spotting system. Liu et al. (2018c), Busta et al. (2017) and He et al. (2018) develop unified text detection and recognition systems with a very similar overall architectures which consist of a detection branch and a recognition branch. Liu et al. (2018c) and Busta et al. (2017) adopt EAST (Zhou et al. 2017) and YOLOv2 (Redmon and Farhadi 2017) as their detection branches respectively, and have a similar text recognition branch in which text proposals are pooled into fixed height tensors by bilinear sampling and then transcribe into strings by a CTC-based recognition module. He et al. (2018) also adopt EAST (Zhou et al. 2017) to generate text proposals, and they introduced character spatial information as explicit supervision in the attention-based recognition branch. Lyu et al. (2018a) propose a modification of Mask R-CNN. For each region of interest, character segmentation maps are produced, indicating the existence and location of a single character. A post-processing step that orders these character from left to right gives the final results. In contrast to the aforementioned works that perform RoI pooling based on oriented bounding boxes, Qin et al. (2019) propose to use axis-aligned bounding boxes and mask the cropped features with a 0/1 textness segmentation mask (He et al. 2017b).

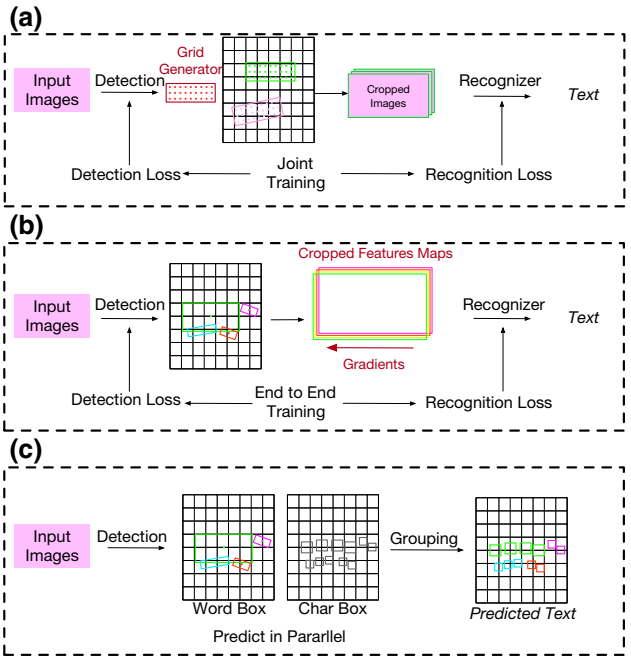

Fig. 8 Illustration of mainstream end-to-end frameworks. **a** In SEE (Bartz et al. 2017), the detection results are represented as grid matrices. Image regions are cropped and transformed before being fed into the recognition branch. **b** Some methods crop from the feature maps and feed them to the recognition branch. **c** While **a**, **b** utilize CTC-based and attention-based recognition branch, it is also possible to retrieve each character as generic objects and compose the text

One-Stage Pipeline In addition to two-staged methods, Xing et al. (2019) predict character and text bounding boxes as well as character type segmentation maps in parallel. The text bounding boxes are then used to group character boxes to form the final word transcription results. This is the first one-staged method.

3.4 Auxiliary Techniques

Recent advances are not limited to detection and recognition models that aim to solve the tasks directly. We should also give credit to auxiliary techniques that have played an important role.

3.4.1 Synthetic Data

Most deep learning models are data-thirsty. Their performance is guaranteed only when enough data are available. In the field of text detection and recognition, this problem is more urgent since most human-labeled datasets are small, usually containing around merely 1K–2K data instances. Fortunately, there have been work (Jaderberg et al. 2014b; Gupta et al. 2016; Zhan et al. 2018; Liao et al. 2019a) that generate data of relatively high quality, and have been widely used for pre-training models for better performance.

Jaderberg et al. (2014b) propose to generate synthetic data for text recognition. Their method blends text with randomly cropped natural images from human-labeled datasets after rending of font, border/shadow, color, and distortion. The results show that training merely on these synthetic data can achieve state-of-the-art performance and that synthetic data can act as augmentative data sources for all datasets.

SynthText (Gupta et al. 2016) first propose to embed text in natural scene images for the training of text detection, while most previous work only print text on a cropped region and these synthetic data are only for text recognition. Printing text on the whole natural images poses new challenges, as it needs to maintain semantic coherence. To produce more realistic data, SynthText makes use of depth prediction (Liu et al. 2015) and semantic segmentation (Arbelaez et al. 2011). Semantic segmentation groups pixels together as semantic clusters and each text instance is printed on one semantic surface, not overlapping multiple ones. A dense depth map is further used to determine the orientation and distortion of the text instance. The model trained only on SynthText achieves state-of-the-art on many text detection datasets. It is later used in other works (Zhou et al. 2017; Shi et al. 2017a) as well for initial pre-training.

Further, Zhan et al. (2018) equip text synthesis with other deep learning techniques to produce more realistic samples. They introduce selective semantic segmentation so that word instances would only appear on sensible objects, e.g. a desk or wall in stead of someone's face. Text rendering in their work is adapted to the image so that they fit into the artistic styles and do not stand out awkwardly.

SynthText3D (Liao et al. 2019a) uses the famous open-source game engine, Unreal Engine 4 (UE4), and UnrealCV (Qiu et al. 2017) to synthesize scene text images. Text is rendered with the scene together and thus can achieve different lighting conditions, weather, and natural occlusions. However, SynthText3D simply follows the pipeline of SynthText and only makes use of the ground-truth depth and segmentation maps provided by the game engine. As a result, SynthText3D relies on manual selection of camera views, which limits its scalability. Besides, the proposed text regions are generated by clipping maximal rectangular bounding boxes extracted from segmentation maps, and therefore are limited to the middle parts of large and well-defined regions, which is an unfavorable location bias.

UnrealText (Long and Yao 2020) is another work using game engines to synthesize scene text images. It features deep interactions with the 3D worlds during synthesis. A ray-casting based algorithm is proposed to navigate in the 3D worlds efficiently and is able to generate diverse camera views automatically. The text region proposal module is based on collision detection and can put text onto the whole surfaces, thus getting rid of the location bias. UnrealText achieves significant speedup and better detector performances.

Text Editing It is also worthwhile to mention the text editing task that is proposed recently (Wu et al. 2019; Yang et al. 2020). Both works try to replace the text content while retaining text styles in natural images, such as the spatial arrangement of characters, text fonts, and colors. Text editing per se is useful in applications such as instant translation using cellphone cameras. It also has great potential in augmenting existing scene text images, though we have not seen any relevant experiment results yet.

3.4.2 Weakly and Semi-supervision

Bootstrapping for Character-Box

Character level annotations are more accurate and better. However, most existing datasets do not provide character-level annotating. Since characters are smaller and close to each other, character-level annotation is more costly and inconvenient. There has been some work on semi-supervised character detection. The basic idea is to initialize a character–detector and applies rules or threshold to pick the most reliable predicted candidates. These reliable candidates are then used as additional supervision sources to refine the character–detector. Both of them aim to augment existing datasets with character level annotations. Their difference is illustrated in Fig. 9.

WordSup (Hu et al. 2017) first initializes the character detector by training 5K warm-up iterations on synthetic datasets. For each image, WordSup generates character candidates, which are then filtered with word-boxes. For characters in each word box, the following score is computed to select the most possible character list:

$$s = w \cdot \frac{area(B_{chars})}{area(B_{word})} + (1 - w) \cdot (1 - \frac{\lambda_2}{\lambda_1}) \quad (1)$$

where B_{chars} is the union of the selected character boxes; B_{word} is the enclosing word bounding box; λ_1 and λ_2 are the first- and second-largest eigenvalues of a covariance matrix C, computed by the coordinates of the centers of the selected character boxes; w is a weight scalar. Intuitively, the first term

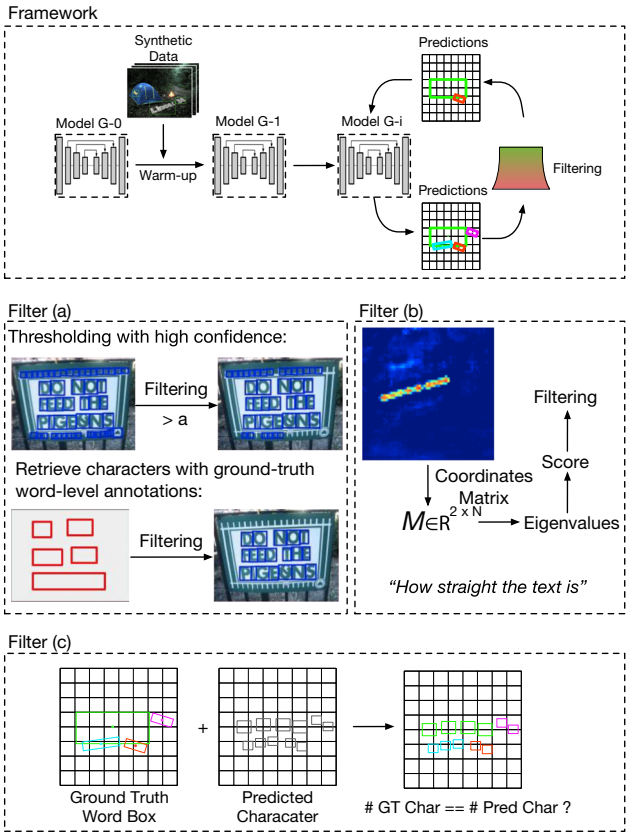

Fig. 9 Overview of semi-supervised and weakly-supervised methods. Existing methods differ in the way with regard to how filtering is done. **a** WeText (Tian et al. 2017), mainly by thresholding the confidence level and filtering by word-level annotation. **b** Scoring-based methods, including WordSup (Hu et al. 2017) which assumes that text are straight lines, and uses a eigenvalue-based metric to measure its *straightness*. **c** by grouping characters into word using ground truth word bounding boxes, and comparing the number of characters (Baek et al. 2019b; Xing et al. 2019).

measures how complete the selected characters can cover the word boxes, while the second term measures whether the selected characters are located on a straight line, which is the main characteristic for word instances in most datasets.

WeText (Tian et al. 2017) start with a small dataset annotated on the character level. It follows two paradigms of bootstrapping: semi-supervised learning and weakly-supervised learning. In the semi-supervised setting, detected character candidates are filtered with a high thresholding value. In the weakly-supervised setting, ground-truth word boxes are used to mask out false positives outside. New instances detected in either way are added to the initial small datasets and re-train the model.

In Baek et al. (2019b) and Xing et al. (2019), the character candidates are filtered with the help of word-level annotations. For each word instance, if the number of detected character bounding boxes inside the word bounding box equals to the length of the ground truth word, the character bounding boxes are regarded as correct.

Partial Annotations In order to improve the recognition performance of end-to-end word spotting models on curved text, Qin et al. (2019) propose to use off-the-shelf straight scene text spotting models to annotate a large number of unlabeled images. These images are called *partially* labeled images, since the off-the-shelf models may omit some words. These partially annotated straight text prove to boost the performance on irregular text greatly.

Another similar effort is the large dataset proposed by Sun et al. (2019), where each image is only annotated with one dominant text. They also design an algorithm to utilize these partially labeled data, which they claim are cheaper to annotate.

4 Benchmark Datasets and Evaluation Protocols

As cutting edge algorithms achieved better performance on existing datasets, researchers are able to tackle more challenging aspects of the problems. New datasets aimed at different real-world challenges have been and are being crafted, benefiting the development of detection and recognition methods further.

In this section, we list and briefly introduce the existing datasets and the corresponding evaluation protocols. We also identify current state-of-the-art approaches to the widely used datasets when applicable.

4.1 Benchmark Datasets

We collect existing datasets and summarize their statistics in Table 1. We select some representative image samples from some of the datasets, which are demonstrated in Fig. 10. Links to these datasets are also collected in our Github repository mentioned in *abstract*, for readers' convenience. In this section, we select some representative datasets and discuss their characteristics.

The ICDAR 2015 incidental text focuses on small and oriented text. The images are taken by Google Glasses without taking care of the image quality. A large proportion of text in the images are very small, blurred, occluded, and multi-oriented, making it very challenging.

The ICDAR MLT 2017 and 2019 datasets contain scripts of 9 and 10 languages respectively. They are the only multi-lingual datasets so far.

Total-Text has a large proportion of curved text, while previous datasets contain only few. These images are mainly taken from street billboards, and annotated as polygons with a variable number of vertices.

Table 1 Public datasets for scene text detection and recognition

Dataset (year)	Image Num (train/val/test)	Orientation	Language	Features	Det.	Recog.
SVT (2010)	100/0/250	Horizontal	EN	–	✓	✓
ICDAR 2003	258/0/251	Horizontal	EN	–	✓	✓
ICDAR 2013	229/0/233	Horizontal	EN	Stroke labels	✓	✓
CUTE (2014)	0/0/80	Curved	EN	–	✓	✓
ICDAR 2015	1000/0/500	Multi-oriented	EN	Blur, small	✓	✓
ICDAR RCTW 2017	8034/0/4229	Multi-oriented	CN	–	✓	✓
Total-Text (2017)	1255/0/300	Curved	EN, CN	Polygon label	✓	✓
CTW (2017)	25000/0/6000	Multi-oriented	CN	Detailed attributes	✓	✓
COCO-Text (2017)	43686/10000/10000	Multi-oriented	En	–	✓	✓
ICDAR MLT 2017	7200/1800/9000	Multi-oriented	9 langanges	–	✓	✓
ICDAR MLT 2019	10000/0/10000	Multi-oriented	10 langanges	–	✓	✓
ICDAR ArT (2019)	5603/0/4563	Curved	EN, CN	–	✓	✓
LSVT (2019)	20157/4968/4841	Multi-oriented	CN	400K partially labeled images	✓	✓
MSRA-TD 500 (2012)	300/0/200	Multi-oriented	EN, CN	Long text	✓	–
HUST-TR 400 (2014)	400/0/0	Multi-oriented	EN, CN	Long text	✓	–
CTW1500 (2017)	1000/0/500	Curved	EN	–	✓	–
SVHN (2010)	73257/0/26032	Horizontal	Digits	Household numbers	–	✓
IIIT5K-Word (2012)	2000/0/3000	Horizontal	EN	–	–	✓
SVTP (2013)	0/0/639	Multi-oriented	EN	Perspective text	–	✓

EN stands for English and *CN* stands for Chinese. Note that HUST-TR 400 is a supplementary training dataset for MSRA-TD 500. *ICDAR 2013* refers to ICDAR 2013 Focused Scene Text Competition. *ICDAR 2015* refers to ICDAR 2015 Incidental Text Competition. The last two columns indicate whether the datasets provide annotations for detection and recognition tasks

Fig. 10 Selected samples from *Chars74K*, *SVT-P*, *IIIT5K*, *MSRA-TD 500*, *ICDAR 2013*, *ICDAR 2015*, *ICDAR 2017 MLT*, *ICDAR 2017 RCTW*, and *Total-Text*

The Chinese Text in the Wild (CTW) dataset (Yuan et al. 2018) contains 32,285 high-resolution street view images, annotated at the character level, including its underlying character type, bounding box, and detailed attributes such as whether it uses *word-art*. The dataset is the largest one to date and the only one that contains detailed annotations. However, it only provides annotations for Chinese text and ignores other scripts, e.g. English.

LSVT (Sun et al. 2019) is composed of two datasets. One is fully labeled with word bounding boxes and word content. The other, while much larger, is only annotated with the word content of the dominant text instance. The authors propose to work on such partially labeled data that are much cheaper.

IIIT 5K-Word (Mishra et al. 2012) is the largest scene text recognition dataset, containing both digital and natural scene images. Its variance in font, color, size, and other noises makes it the most challenging one to date.

4.2 Evaluation Protocols

In this part, we briefly summarize the evaluation protocols for text detection and recognition.

As metrics for performance comparison of different algorithms, we usually refer to their precision, recall and F1-score. To compute these performance indicators, the list of predicted text instances should be matched to the ground truth labels in the first place. Precision, denoted as P, is calculated as the proportion of predicted text instances that can be matched to ground truth labels. Recall, denoted as R, is the proportion of ground truth labels that have correspondents in the predicted list. F1-score is then computed by $F_1 = \frac{2*P*R}{P+R}$, taking both precision and recall into account. Note that the matching between the predicted instances and ground truth ones comes first.

4.2.1 Text Detection

There are mainly two different protocols for text detection, the IOU based PASCAL Eval and overlap based DetEval. They differ in the criterion of matching predicted text instances and ground truth ones. In the following part, we use these notations: S_{GT} is the area of the ground truth bounding box, S_P is the area of the predicted bounding box, S_I is the area of the intersection of the predicted and ground truth bounding box, S_U is the area of the union.

- *DetEval* DetEval imposes constraints on both precision, i.e. $\frac{S_I}{S_P}$ and recall, i.e. $\frac{S_I}{S_{GT}}$. Only when both are larger than their respective thresholds, are they matched together.
- *PASCAL* (Everingham et al. 2015) The basic idea is that, if the intersection-over-union value, i.e. $\frac{S_I}{S_U}$, is larger than a designated threshold, the predicted and ground truth box are matched together.

Most works follow either one of the two evaluation protocols, but with small modifications. We only discuss those that are different from the two protocols mentioned above.

- *ICDAR-2003/2005* The match score m is calculated in a way similar to IOU. It is defined as the ratio of the area of intersection over that of the minimum bounding rectangular bounding box containing both.
- *ICDAR-2011/2013* One major drawback of the evaluation protocol of ICDAR2003/2005 is that it only considers the one-to-one match. It does not consider

one-to-many, many-to-many, and many-to-one matching, which underestimates the actual performance. Therefore, ICDAR-2011/2013 follows the method proposed by Wolf and Jolion (2006), where one-to-one matching is assigned a score of 1 and the other two types are punished to a constant score less than 1, usually set as 0.8.

- *MSRA-TD 500* (Tu et al. 2012) propose a new evaluation protocol for rotated bounding boxes, where both the predicted and ground truth bounding box are revolved horizontally around its center. They are matched only when the standard IOU score is higher than the threshold and the rotation of the original bounding boxes is less a pre-defined value (in practice $\frac{pi}{4}$).
- *TIoU* (Liu et al. 2019) Tightness-IoU takes into account the fact that scene text recognition is sensitive to missing parts and superfluous parts in detection results. Not-retrieved areas will result in missing characters in recognition results, and redundant areas will result in unexpected characters. The proposed metrics penalize IoUs by scaling it down by the proportion of missing areas and the proportion of superfluous areas that overlap with other text.

The main drawback of existing evaluation protocols is that they only consider the best F1 scores under arbitrarily selected confidence thresholds selected on test sets. Qin et al. (2019) also evaluate their method with the average precision (AP) metric that is widely adopted in general object detection. While F1 scores are only single points on the precision-recall curves, AP values consider the whole precision-recall curves. Therefore, AP is a more comprehensive metric and we urge that researchers in this field use AP instead of F1 alone.

4.2.2 Text Recognition and End-to-End System

In scene text recognition, the predicted text string is compared to the ground truth directly. The performance evaluation is in either character-level recognition rate (i.e. how many characters are recognized) or word level (whether the predicted word exactly the same as ground truth). ICDAR also introduces an edit-distance based performance evaluation.

In end-to-end evaluation, matching is first performed in a similar way to that of text detection, and then the text content is compared.

The most widely used datasets for end-to-end systems are ICDAR 2013 (Karatzas et al. 2013) and ICDAR 2015 (Karatzas et al. 2015). The evaluation over these two datasets are carried out under two different settings,[2] the *Word Spotting* setting and the *End-to-End* setting. Under *Word Spotting*, the performance evaluation only focuses on the text instances from the scene image that appears in a predesignated vocabulary, while other text instances are ignored. On the contrary, all text instances that appear in the scene image are included under *End-to-End*. Three different vocabulary lists are provided for candidate transcriptions. They include *Strongly Contextualised*, *Weakly Contextualised*, and *Generic*. The three kinds of lists are summarized in Table 8. Note that under *End-to-End*, these vocabularies can still serve as references.

[2] http://rrc.cvc.uab.es/files/Robust_Reading_2015_v02.pdf.

Table 2 Detection on ICDAR 2013

Method	P	R	F1
Zhang et al. (2016)	88	78	83
Gupta et al. (2016)	92.0	75.5	83.0
Yao et al. (2016)	88.88	80.22	84.33
Deng et al. (2018)	86.4	83.6	84.5
He et al. (2017a)(∗)	93	79	85
Shi et al. (2017a)	87.7	83.0	85.3
Lyu et al. (2018b)	93.3	79.4	85.8
He et al. (2017d)	92	80	86
Liao et al. (2017)	89	83	86
Zhou et al. (2017)	92.64	82.67	87.37
Liu et al. (2018e)	88.2	87.2	87.7
Tian et al. (2016)	93	83	88
He et al. (2017c)	89	86	88
He et al. (2018)	88	87	88
Xue et al. (2018)	91.5	87.1	89.2
Hu et al. (2017)(∗)	93.34	87.53	90.34
Lyu et al. (2018a)(∗)	94.1	88.1	91.0
Zhang et al. (2018)	93.7	90.0	92.3
Baek et al. (2019b)	97.4	93.1	95.2

Evaluation results of recent methods on several widely adopted benchmark datasets are summarized in the following tables: Table 2 for detection on ICDAR 2013, Table 4 for detection on ICDAR 2015 Incidental Text, Table 3 for detection on ICDAR 2017 MLT, Table 5 for detection and end-to-end word spotting on Total-Text, Table 6 for detection on CTW1500, Table 7 for detection on MSRA-TD 500, Table 9 for recognition on several datasets, and Table 10 for end-to-end text spotting on ICDAR 2013 and ICDAR 2015. Note that, we do not report performance under multi-scale conditions if single-scale performances are reported. We use ∗ to indicate methods where only multi-scale performances are reported. Since different backbone feature extractors are used in some works, we only report performances based on ResNet-50 unless not provided. For a better illustration, we plot the recent progress of detection performance in Fig. 11, and recognition performance in Fig. 12.

Note that, current evaluation for scene text recognition may be problematic. According to Baek et al. (2019a), most

Table 3 Detection on ICDAR MLT 2017

Method	P	R	F1
Liu et al. (2018c)	81.0	57.5	67.3
Zhang et al. (2019)	60.6	78.8	68.5
Wang et al. (2019a)	73.4	69.2	72.1
Xing et al. (2019)	70.10	77.07	73.42
Baek et al. (2019b)	68.2	80.6	73.9
Long and Yao (2020)	82.2	67.4	74.1

Table 4 Detection on ICDAR 2015

Method	P	R	F1	FPS
Zhang et al. (2016)	71	43.0	54	0.5
Tian et al. (2016)	74	52	61	–
He et al. (2017a)(∗)	76	54	63	–
Yao et al. (2016)	72.26	58.69	64.77	1.6
Shi et al. (2017a)	73.1	76.8	75.0	–
Liu et al. (2018e)	72	80	76	–
He et al. (2017c)	80	73	77	7.7
Hu et al. (2017)(∗)	79.33	77.03	78.16	2.0
Zhou et al. (2017)	83.57	73.47	78.20	13.2
Wang et al. (2018)	85.7	74.1	79.5	–
Lyu et al. (2018b)	94.1	70.7	80.7	3.6
He et al. (2017d)	82	80	81	–
Jiang et al. (2017)	85.62	79.68	82.54	–
Long et al. (2018)	84.9	80.4	82.6	10.2
He et al. (2018)	84	83	83	1.1
Lyu et al. (2018a)	85.8	81.2	83.4	4.8
Deng et al. (2018)	85.5	82.0	83.7	3.0
Zhang et al. (2020)	88.53	84.69	86.56	–
Wang et al. (2019a)	86.92	84.50	85.69	1.6
Tian et al. (2019)	88.3	85.0	86.6	3
Baek et al. (2019b)	89.8	84.3	86.9	8.6
Zhang et al. (2019)	83.5	91.3	87.2	–
Qin et al. (2019)	89.36	85.75	87.52	4.76
Wang et al. (2019b)	89.2	86.0	87.6	10.0
Xing et al. (2019)	88.30	91.15	89.70	–

researchers are actually using different subsets when they refer to the same dataset, causing discrepancies in performance. Besides, Long and Yao (2020) further point out that half of the widely adopted benchmark datasets have imperfect annotations, e.g. ignoring case-sensitivities and punctuations, and provide new annotations for those datasets. Though most paper claim to train their models to recognize in a case-sensitive way and also include punctuations, they may be limiting their output to only digits and case-insensitive characters during evaluation.

Table 5 Detection and end-to-end on total-text

Method	Detection			E2E
	P	R	F	
Lyu et al. (2018a)	69.0	55.0	61.3	52.9
Long et al. (2018)	82.7	74.5	78.4	–
Wang et al. (2019b)	80.9	76.2	78.5	–
Wang et al. (2019a)	84.02	77.96	80.87	–
Zhang et al. (2019)	75.7	88.6	81.6	–
Baek et al. (2019b)	87.6	79.9	83.6	–
Qin et al. (2019)	83.3	83.4	83.3	67.8
Xing et al. (2019)	81.0	88.6	84.6	63.6
Zhang et al. (2020)	86.54	84.93	85.73	–

Table 6 Detection on CTW1500

Method	P	R	F1
Liu et al. (2017)	77.4	69.8	73.4
Long et al. (2018)	67.9	85.3	75.6
Zhang et al. (2019)	69.6	89.2	78.4
Wang et al. (2019b)	80.1	80.2	80.1
Tian et al. (2019)	82.7	77.8	80.1
Wang et al. (2019a)	84.84	79.73	82.2
Baek et al. (2019b)	86.0	81.1	83.5
Zhang et al. (2020)	85.93	83.02	84.45

Table 7 Detection on MSRA-TD 500

Method	P	R	F1
Kang et al. (2014)	71	62	66
Zhang et al. (2016)	83	67	74
He et al. (2017d)	77	70	74
Yao et al. (2016)	76.51	75.31	75.91
Zhou et al. (2017)	87.28	67.43	76.08
Wu and Natarajan (2017)	77	78	77
Shi et al. (2017a)	86	70	77
Deng et al. (2018)	83.0	73.2	77.8
Long et al. (2018)	83.2	73.9	78.3
Xue et al. (2018)	83.0	77.4	80.1
Wang et al. (2018)	90.3	72.3	80.3
Lyu et al. (2018b)	87.6	76.2	81.5
Baek et al. (2019b)	88.2	78.2	82.9
Tian et al. (2019)	84.2	81.7	82.9
Liu et al. (2018e)	88	79	83
Wang et al. (2019b)	85.2	82.1	83.6
Zhang et al. (2020)	88.05	82.30	85.08

Table 8 Characteristics of the three vocabulary lists used in ICDAR 2013/2015

Vocab list	Description
S	Per-image list of 100 words including all words in the image
W	All words in the entire test set
G	A 90k-word generic vocabulary

S stands for *Strongly Contextualised*, *W* for *Weakly Contextualised*, and *G* for *Generic*

5 Application

The detection and recognition of text—the visual and physical carrier of human civilization—allow the connection between vision and the understanding of its content further. Apart from the applications we have mentioned at the beginning of this paper, there have been numerous specific application scenarios across various industries and in our daily lives. In this part, we list and analyze the most outstanding ones that have, or are to have, significant impact, improving our productivity and life quality.

Automatic Data Entry Apart from an electronic archive of existing documents, OCR can also improve our productivity in the form of automatic data entry. Some industries involve time-consuming data type-in, e.g. express orders written by customers in the delivery industry, and hand-written information sheets in the financial and insurance industries. Applying OCR techniques can accelerate the data entry process as well as protect customer privacy. Some companies have already been using these technologies, e.g. SF-Express.[3] Another potential application is *note taking*, such as NEBO,[4] a note-taking software on tablets like iPad that performs instant transcription as users write down notes.

Identity Authentication Automatic identity authentication is yet another field where OCR can give a full play to. In fields such as Internet finance and Customs, users/passengers are required to provide identification (ID) information, such as identity card and passport. Automatic recognition and analysis of the provided documents would require OCR that reads and extracts the textual content, and can automate and greatly accelerate such processes. There are companies that have already started working on identification based on face and ID card, e.g. MEGVII (Face++).[5]

Augmented Computer Vision As text is an essential element for the understanding of scene, OCR can assist computer vision in many ways. In the scenario of autonomous vehicles, text-embedded panels carry important information, e.g. geo-location, current traffic condition, navigation, and etc.. There have been several works on text detection and recognition for autonomous vehicle (Mammeri et al. 2014, 2016). The largest dataset so far, CTW (Yuan et al. 2018), also places extra emphasis on traffic signs. Another example is the instant translation, where OCR is combined with a translation model. This is extremely helpful and time-saving as people travel or read documents written in foreign languages. Google's Translate application[6] can perform such instant translation. A similar application is instant text-to-speech software equipped with OCR, which can help those with visual disability and those who are illiterate.[7]

Intelligent Content Analysis OCR also allows the industry to perform more intelligent analysis, mainly for platforms like video-sharing websites and e-commerce. Text can be extracted from images and subtitles as well as real-time commentary subtitles (a kind of floating comments added by users, e.g. those in Bilibili[8] and Niconico[9]). On the one hand, such extracted text can be used in automatic content tagging and recommendation systems. They can also be used to perform user sentiment analysis, e.g. which part of the video attracts the users most. On the other hand, website administrators can impose supervision and filtration for inappropriate and illegal content, such as terrorist advocacy.

6 Conclusion and Discussion

6.1 Status Quo

Algorithms The past several years have witnessed the significant development of algorithms for text detection and recognition, mainly due to the deep learning boom. Deep learning models have replaced the manual search and design for patterns and features. With the improved capability of models, research attention has been drawn to challenges such as oriented and curved text detection, and have achieved considerable progress.

Applications Apart from efforts towards a general solution to all sorts of images, these algorithms can be trained and adapted to more specific scenarios, e.g. *bankcard*, *ID card*, and *driver's license*. Some companies have been providing such scenario-specific APIs, including Baidu Inc., Tencent Inc., and MEGVII Inc.. Recent development of fast and efficient methods (Ren et al. 2015; Zhou et al. 2017) has also allowed the deployment of large-scale systems (Borisyuk et al. 2018). Companies including Google Inc. and Amazon Inc. are also providing text extraction APIs.

[3] Official website: http://www.sf-express.com/cn/sc/.

[4] Official website: https://www.myscript.com/nebo/.

[5] https://www.faceplusplus.com/face-based-identification/.

[6] https://translate.google.com/.

[7] https://en.wikipedia.org/wiki/Screen_reader#cite_note-Braille_display-2.

[8] https://www.bilibili.com.

[9] www.nicovideo.jp/.

Table 9 State-of-the-art recognition performance across a number of datasets

Methods	ConvNet, Data	IIIT5k			SVT		IC03			IC13	IC15	SVTP	CUTE	Total-text
		50	1k	0	50	0	50	Full	0	0	0	0	0	0
Yao et al. (2014)	–	80.2	69.3	–	75.9	–	88.5	80.3	–	–	–	–	–	–
Jaderberg et al. (2014c)	–	–	–	–	86.1	–	96.2	91.5	–	–	–	–	–	–
Su and Lu (2014)	–	–	–	–	83.0	–	92.0	82.0	–	–	–	–	–	–
Gordo (2015)	–	93.3	86.6	–	91.8	–	–	–	–	–	–	–	–	–
Jaderberg et al. (2016)	VGG, 90k	97.1	92.7	–	95.4	80.7	98.7	98.6	93.1	90.8	–	–	–	–
Shi et al. (2017b)	VGG, 90k	97.8	95.0	81.2	97.5	82.7	98.7	98.0	91.9	89.6	–	–	–	–
Shi et al. (2016)	VGG, 90k	96.2	93.8	81.9	95.5	81.9	98.3	96.2	90.1	88.6	–	71.8	59.2	–
Lee and Osindero (2016)	VGG, 90k	96.8	94.4	78.4	96.3	80.7	97.9	97.0	88.7	90.0	–	–	–	–
Yang et al. (2017)	VGG, Private	97.8	96.1	–	95.2	–	97.7	–	–	–	–	75.8	69.3	–
Cheng et al. (2017a)	ResNet, 90k +ST^+	99.3	97.5	87.4	97.1	85.9	99.2	97.3	94.2	93.3	70.6	–	–	–
Shi et al. (2018)	ResNet, 90k + ST	99.6	98.8	93.4	99.2	93.6	98.8	98.0	94.5	91.8	76.1	78.5	79.5	–
Liao et al. (2019b)	ResNet, ST^+ + Private	99.8	98.8	91.9	98.8	86.4	–	–	–	91.5	–	–	79.9	–
Li et al. (2019)	ResNet, 90k + ST + Private	–	–	91.5	–	84.5	–	–	–	91.0	69.2	76.4	83.3	–
Zhan and Lu (2019)	ResNet, 90k + ST	99.6	98.8	93.3	97.4	90.2	–	–	–	91.3	76.9	79.6	83.3	–
Yang et al. (2019)	ResNet, 90k + ST	99.5	98.8	94.4	97.2	88.9	99.0	98.3	95.0	93.9	78.7	80.8	87.5	–
Long et al. (2019)	ResNet, 90k + Curved ST	–	–	94.8	–	89.6	–	–	95.8	92.8	78.2	81.6	89.6	76.3
Yu et al. (2020)	ResNet, 90k + ST	–	–	94.8	–	91.5	–	–	–	95.5	82.7	85.1	87.8	–

"50", "1k", "Full" are lexicons. "0" means no lexicon. "90k" and "ST" are the Synth90k and the SynthText datasets, respectively. "ST^+" means including character-level annotations. "Private" means private training data

Table 10 Performance of end-to-end and word spotting on ICDAR 2015 and ICDAR 2013

Method	Word spotting			End-to-end		
	S	W	G	S	W	G
ICDAR 2015						
Liu et al. (2018c)	84.68	79.32	63.29	81.09	75.90	60.80
Xing et al. (2019)	–	–	–	80.14	74.45	62.18
Lyu et al. (2018a)	79.3	74.5	64.2	79.3	73.0	62.4
He et al. (2018)	85	80	65	82	77	63
Qin et al. (2019)	–	–	–	83.38	79.94	67.98
ICDAR 2013						
Busta et al. (2017)	92	89	81	89	86	77
Liu et al. (2018c)	92.73	90.72	83.51	88.81	87.11	80.81
Li et al. (2017a)	94.2	92.4	88.2	91.1	89.8	84.6
He et al. (2018)	93	92	87	91	89	86
Lyu et al. (2018a)	92.5	92.0	88.2	92.2	91.1	86.5

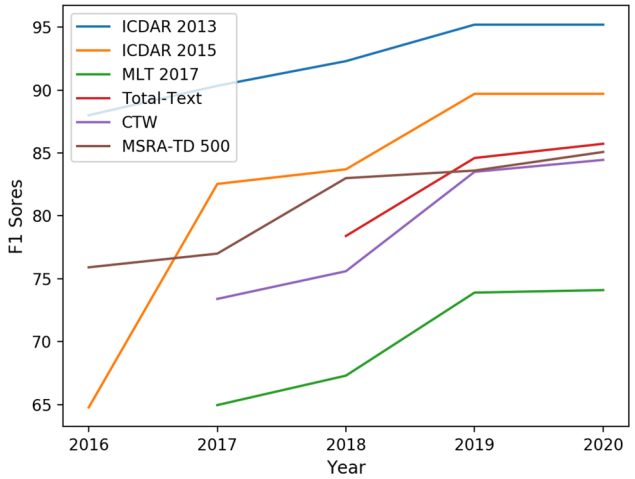

Fig. 11 Progress of scene text detection over the past few years (evaluated as F1 scores)

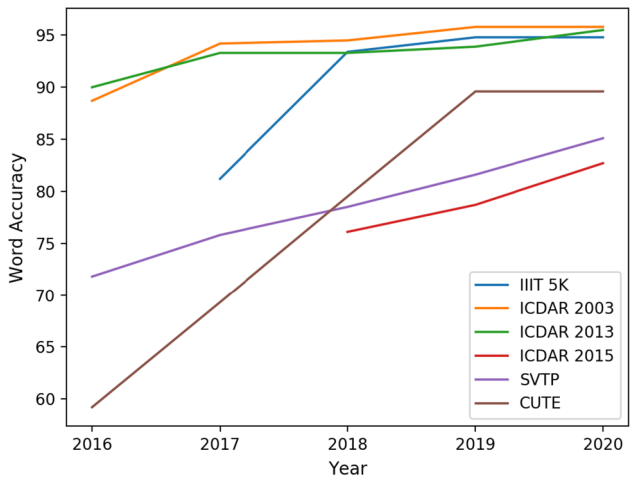

Fig. 12 Progress of scene text recognition over the past few years (evaluated as word-level accuracy)

6.2 Challenges and Future Trends

We look at the present through a rear-view mirror. We march backward into the future (Liu 1975). We list and discuss challenges, and analyze what would be the next valuable research directions in the field scene text detection and recognition.

Languages There are more than 1000 languages in the world.[10] However, most current algorithms and datasets have primarily focused on text of English. While English has a rather small alphabet, other languages such as Chinese and Japanese have a much larger one, with tens of thousands of symbols. RNN-based recognizers may suffer from such enlarged symbol sets. Moreover, some languages have much more complex appearances, and they are therefore more sensitive to conditions such as image quality. Researchers should first verify how well current algorithms can generalize to text of other languages and further to mixed text. Unified detection and recognition systems for multiple types of languages are of important academic value and application prospects. A feasible solution might be to explore compositional representations that can capture the common patterns of text instances of different languages, and train the detection and recognition models with text examples of different languages, which are generated by text synthesizing engines.

Robustness of Models Although current text recognizers have proven to be able to generalize well to different scene text datasets even only using synthetic data, recent work (Liao et al. 2019b) shows that robustness against flawed detection is not a neglectable problem. Actually, such instability in prediction has also been observed for text detection models. The reason behind this kind of phenomenon is still unclear. One conjecture is that the robustness of models is related to the internal operating mechanism of deep neural networks.

[10] https://www.ethnologue.com/guides/how-many-languages.

Generalization Few detection algorithms except for TextSnake (Long et al. 2018) have considered the problem of generalization ability across datasets, i.e. training on one dataset, and testing on another. Generalization ability is important as some application scenarios would require the adaptability to varying environments. For example, instant translation and OCR in autonomous vehicles should be able to perform stably under different situations: zoomed-in images with large text instances, far and small words, blurred words, different languages, and shapes. It remains unverified whether simply pooling all existing datasets together is enough, especially when the target domain is totally unknown.

Evaluation Existing evaluation metrics for detection stem from those for general object detection. Matching based on IoU score or pixel-level precision and recall ignore the fact that *missing parts* and *superfluous backgrounds* may hurt the performance of the subsequent recognition procedure. For each text instance, pixel-level precision and recall are good metrics. However, their scores are assigned to 1.0 once they are matched to ground truth, and thus not reflected in the final dataset-level score. An off-the-shelf alternative method is to simply sum up the instance-level scores under DetEval instead of first assigning them to 1.0.

Synthetic Data While training recognizers on synthetic datasets has become a routine and results are excellent, detectors still rely heavily on real datasets. It remains a challenge to synthesize diverse and realistic images to train detectors. Potential benefits of synthetic data are not yet fully explored, such as generalization ability. Synthesis using 3D engines and models can simulate different conditions such as lighting and occlusion, and thus is worth further development.

Efficiency Another shortcoming of deep-learning-based methods lies in their efficiency. Most of the current systems can not run in real-time when deployed on computers without GPUs or mobile devices. Apart from model compression and lightweight models that have proven effective in other tasks, it is also valuable to study how to make custom speedup mechanism for text-related tasks.

Bigger and Better Datasets The sizes of most widely adopted datasets are small ($\sim$1k images). It will be worthwhile to study whether the improvements gained from current algorithms can scale up or they are just accidental results of better regularization. Besides, most datasets are only labeled with bounding boxes and texts. Detailed annotation of different attributes (Yuan et al. 2018) such as *word-art* and *occlusion* may guide researchers with pertinence. Finally, datasets characterized by real-world challenges are also important in advancing research progress, such as densely located text on products. Another related problem is that most of the existing datasets do not have validation sets. It is highly possible that the current reported evaluation results are actually upward biased due to overfitting on the test sets. We suggest that researchers should focus on large datasets, such as ICDAR MLT 2017, ICDAR MLT 2019, ICDAR ArT 2019, and COCO-Text.

References

Almazán, J., Gordo, A., Fornés, A., & Valveny, E. (2014). Word spotting and recognition with embedded attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(12), 2552–2566.

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 898–916.

Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., et al. (2019a). What is wrong with scene text recognition model comparisons? Dataset and model analysis. In *Proceedings of the IEEE international conference on computer vision* (pp. 4715–4723).

Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019b). Character region awareness for text detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 9365–9374).

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Bai, F., Cheng, Z., Niu, Y., Pu, S., & Zhou, S. (2018). Edit probability for scene text recognition. In *CVPR 2018*.

Bartz, C., Yang, H., & Meinel, C. (2017). See: Towards semi-supervised end-to-end scene text recognition. arXiv preprint arXiv:1712.05404.

Bissacco, A., Cummins, M., Netzer, Y., & Neven, H. (2013). Photoocr: Reading text in uncontrolled conditions. In *Proceedings of the IEEE international conference on computer vision* (pp. 785–792).

Borisyuk, F., Gordo, A., & Sivakumar, V. (2018). Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 71–79). ACM.

Busta, M., Neumann, L., & Matas, J. (2015). Fastext: Efficient unconstrained scene text detector. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 1206–1214).

Busta, M., Neumann, L., & Matas, J. (2017). Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *Proceedings of ICCV*.

Chen, X., Yang, J., Zhang, J., & Waibel, A. (2004). Automatic detection and recognition of signs from natural scenes. *IEEE Transactions on Image Processing*, *13*(1), 87–99.

Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., & Zhou, S. (2017a). Focusing attention: Towards accurate text recognition in natural images. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 5086–5094). IEEE.

Cheng, Z., Liu, X., Bai, F., Niu, Y., Pu, S., & Zhou, S. (2017b). Arbitrarily-oriented text recognition. In *CVPR2018*.

Ch'ng, C.K., & Chan, C. S. (2017). Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)* (Vol. 1, pp. 935–942). IEEE.

Chowdhury, M. A., & Deb, K. (2013). Extracting and segmenting container name from container images. *International Journal of Computer Applications*, *74*(19), 18–22.

Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., et al. (2011). Text detection and character recognition in scene images with unsupervised feature learning. In *2011 international conference on document analysis and recognition (ICDAR)* (pp. 440–445). IEEE.

Dai, Y., Huang, Z., Gao, Y., & Chen, K. (2017). Fused text segmentation networks for multi-oriented scene text detection. arXiv preprint arXiv:1709.03272.

Dalal, N., & Triggs, B., (2005). Histograms of oriented gradients for human detection. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. 886–893). IEEE.

Deng, D., Liu, H., Li, X., & Cai, D. (2018). Pixellink: Detecting scene text via instance segmentation. *Proceedings of AAA*, *I*, 2018.

DeSouza, G. N., & Kak, A. C. (2002). Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(2), 237–267.

Dollár, P., Appel, R., Belongie, S., & Perona, P. (2014). Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(8), 1532–1545.

Dvorin, Y., & Havosha, U. E. (2009). Method and device for instant translation, June 4. US Patent App. 11/998,931.

Epshtein, B., Ofek, E., & Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. In *2010 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2963–2970). IEEE.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, *111*(1), 98–136.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*(1), 55–79.

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). DSSD: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659.

Gao, Y., Chen, Y., Wang, J., & Lu, H. (2017). Reading scene text with attention convolutional sequence modeling. arXiv preprint arXiv:1709.04303.

Girshick, R. (2015). Fast R-CNN. In *The IEEE international conference on computer vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 580–587).

Goldberg, A. V. (1997). An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms*, *22*(1), 1–29.

Gordo, A. (2015). Supervised mid-level features for word image representation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2956–2964).

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on machine learning* (pp. 369–376). ACM.

Graves, A., Liwicki, M., Bunke, H., Schmidhuber, J., & Fernández, S. (2008). Unconstrained on-line handwriting recognition with recurrent neural networks. In *Advances in neural information processing systems* (pp. 577–584).

Gupta, A., Vedaldi, A., & Zisserman, A. (2016). Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2315–2324).

Ham, Y. K., Kang, M. S., Chung, H. K., Park, R.-H., & Park, G. T. (1995). Recognition of raised characters for automatic classification of rubber tires. *Optical Engineering*, *34*(1), 102–110.

Han, J., Zhang, D., Cheng, G., Liu, N., & Xu, D. (2018). Advanced deep-learning techniques for salient and category-specific object detection: A survey. *IEEE Signal Processing Magazine*, *35*(1), 84–100.

He, D., Yang, X., Liang, C., Zhou, Z., Ororbia, A. G., Kifer, D., & Giles, C. L. (2017a). Multi-scale FCN with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 474–483). IEEE.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017b). Mask R-CNN. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 2980–2988). IEEE.

He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., & Li, X. (2017c). Single shot text detector with regional attention. In *The IEEE international conference on computer vision (ICCV)*.

He, P., Huang, W., Qiao, Y., Loy, C. C., & Tang, X. (2016). Reading scene text in deep convolutional sequences. In *Thirtieth AAAI conference on artificial intelligence*.

He, T., Tian, Z., Huang, W., Shen, C., Qiao, Y., & Sun, C. (2018). An end-to-end textspotter with explicit alignment and attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5020–5029).

He, W., Zhang, X.-Y., Yin, F., & Liu, C.-L. (2017d). Deep direct regression for multi-oriented scene text detection. In *The IEEE international conference on computer vision (ICCV)*.

He, Z., Liu, J., Ma, H., & Li, P. (2005). A new automatic extraction method of container identity codes. *IEEE Transactions on Intelligent Transportation Systems*, *6*(1), 72–78.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hu, H., Zhang, C., Luo, Y., Wang, Y., Han, J., & Ding, E. (2017). Wordsup: Exploiting word annotations for character based text detection. In *Proceedings of the IEEE international conference on computer vision. 2017*.

Huang, W., Lin, Z., Yang, J., & Wang, J. (2013). Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proceedings of the IEEE international conference on computer vision* (pp. 1241–1248).

Huang, W., Qiao, Y., & Tang, X. (2014). Robust scene text detection with convolution neural network induced MSER trees. In *European conference on computer vision* (pp. 497–511). Springer.

Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014a). Deep structured output learning for unconstrained text recognition. In *ICLR2015*.

Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014b). Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227.

Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2016). Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, *116*(1), 1–20.

Jaderberg, M., Simonyan, K., Zisserman, A. et al. (2015). Spatial transformer networks. In *Advances in neural information processing systems* (pp. 2017–2025).

Jaderberg, M., Vedaldi, A., & Zisserman, A. (2014c). Deep features for text spotting. In *In Proceedings of European conference on computer vision (ECCV)* (pp. 512–528). Springer.

Jain, A. K., & Yu, B. (1998). Automatic text location in images and video frames. *Pattern Recognition*, *31*(12), 2055–2076.

Jiang, Y., Zhu, X., Wang, X., Yang, S., Li, W., Wang, H., Fu, P., & Luo, Z. (2017). R2CNN: rotational region CNN for orientation robust scene text detection. arXiv preprint arXiv:1706.09579.

Jung, K., Kim, K. I., & Jain, A. K. (2004). Text information extraction in images and video: A survey. *Pattern Recognition*, *37*(5), 977–997.

Kang, L., Li, Y., & Doermann, D. (2014). Orientation robust text line detection in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4034–4041).

Karatzas, D., & Antonacopoulos, A. (2004). Text extraction from web images based on a split-and-merge segmentation method using colour perception. In *Proceedings of the 17th international conference on pattern recognition, 2004. ICPR 2004* (Vol. 2, pp. 634–637). IEEE.

Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., et al. (2015). ICDAR 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)* (pp. 1156–1160). IEEE.

Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L. G. I., Mestre, S. R., et al. (2013). ICDAR 2013 robust reading competition. In *2013 12th international conference on document analysis and recognition (ICDAR)* (pp. 1484–1493). IEEE.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lee, C.-Y., & Osindero, S. (2016). Recursive recurrent nets with attention modeling for OCR in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2231–2239).

Lee, J.-J, Lee, P.-H., Lee, S.-W., Yuille, A., & Koch, C. (2011). Adaboost for text detection in natural scene. In *2011 international conference on document analysis and recognition (ICDAR)* (pp. 429–434). IEEE.

Lee, S., & Kim, J. H. (2013). Integrating multiple character proposals for robust scene text extraction. *Image and Vision Computing*, *31*(11), 823–840.

Li, H., Wang, P., & Shen, C. (2017a). Towards end-to-end text spotting with convolutional recurrent neural networks. In *The IEEE international conference on computer vision (ICCV)*.

Li, H., Wang, P., Shen, C., & Zhang, G. (2019). Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*.

Li, R., En, M., Li, J., & Zhang, H. (2017b). Weakly supervised text attention network for generating text proposals in scene images. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)* (Vol. 1, pp. 324–330). IEEE.

Liao, M., Shi, B., & Bai, X. (2018a). Textboxes++: A single-shot oriented scene text detector. *IEEE Transactions on Image Processing*, *27*(8), 3676–3690.

Liao, M., Shi, B., Bai, X., Wang, X., & Liu, W. (2017). Textboxes: A fast text detector with a single deep neural network. In *AAAI* (pp. 4161–4167).

Liao, M., Song, B., He, M., Long, S., Yao, C., & Bai, X. (2019a). Synthtext3d: Synthesizing scene text images from 3d virtual worlds. arXiv preprint arXiv:1907.06007.

Liao, M., Zhang, J., Wan, Z., Xie, F., Liang, J., Lyu, P., Yao, C., & Bai, X. (2019b). Scene text recognition from two-dimensional perspective. In *AAAI*.

Liao, M., Zhu, Z., Shi, B., Xia, G.-S., & Bai, X. (2018b). Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5909–5918).

Liu, F., Shen, C., & Lin, G. (2015). Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5162–5170).

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2018a). Deep learning for generic object detection: A survey. arXiv preprint arXiv:1809.02165.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016a). SSD: Single shot multibox detector. In *In Proceedings of European conference on computer vision (ECCV)* (pp. 21–37). Springer.

Liu, W., Chen, C., & Wong, K. (2018b). Char-net: A character-aware neural network for distorted scene text recognition. In *AAAI conference on artificial intelligence*, New Orleans, Louisiana, USA.

Liu, W., Chen, C., Wong, K.-Y. K., Su, Z., & Han, J. (2016b). Star-net: A spatial attention residue network for scene text recognition. In *BMVC* (Vol. 2, p. 7).

Liu, X. (1975). *Old book of tang*. Beijing: Zhonghua Book Company.

Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., & Yan, J. (2018c). FOTS: Fast oriented text spotting with a unified network. In *CVPR2018*.

Liu, X., & Samarabandu, J. (2005a). An edge-based text region extraction algorithm for indoor mobile robot navigation. In *2005 IEEE international conference mechatronics and automation* (Vol. 2, pp. 701–706). IEEE.

Liu, X., & Samarabandu, J. K. (2005b). A simple and fast text localization algorithm for indoor mobile robot navigation. In *Image processing: Algorithms and systems IV* (Vol. 5672, pp. 139–151). International Society for Optics and Photonics.

Liu, Y., & Jin, L. (2017). Deep matching prior network: Toward tighter multi-oriented text detection.

Liu, Y., Jin, L., Xie, Z., Luo, C., Zhang, S., & Xie, L. (2019). Tightness-aware evaluation protocol for scene text detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9612–9620).

Liu, Y., Jin, L., Zhang, S., & Zhang, S. (2017). Detecting curve text in the wild: New dataset and new solution. arXiv preprint arXiv:1712.02170.

Liu, Z., Li, Y., Ren, F., Yu, H., & Goh, W. (2018d). Squeezedtext: A real-time scene text recognition by binary convolutional encoder–decoder network. In *AAAI*.

Liu, Z., Lin, G., Yang, S., Feng, J., Lin, W., & Goh, W. L. (2018e). Learning Markov clustering networks for scene text detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 6936–6944).

Long, S., Guan, Y., Bian, K., & Yao, C. (2020). A new perspective for flexible feature gathering in scene text recognition via character anchor pooling. In *ICASSP 2020—2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 2458–2462. IEEE.

Long, S., Guan, Y., Wang, B., Bian, K., & Yao, C. (2019). Alchemy: Techniques for rectification based irregular scene text recognition. arXiv preprint arXiv:1908.11834.

Long, S., Ruan, J., Zhang, W., He, X., Wu, W., & Yao, C. (2018). Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proceedings of European conference on computer vision (ECCV)*.

Long, S., & Yao, C. (2020). Unrealtext: Synthesizing realistic scene text images from the unreal world. arXiv preprint arXiv:2003.10608.

Lyu, P., Liao, M., Yao, C., Wu, W., & Bai, X. (2018a). Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of European conference on computer vision (ECCV)*.

Lyu, P., Yao, C., Wu, W., Yan, S., & Bai, X. (2018b). Multi-oriented scene text detection via corner localization and region segmentation. In *2018 IEEE conference on computer vision and pattern recognition (CVPR)*.

Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., et al. (2018). Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, *20*, 3111–3122.

Mammeri, A., & Boukerche, A. et al. (2016). MSER-based text detection and communication algorithm for autonomous vehicles. In *2016 IEEE symposium on computers and communication (ISCC)* (pp. 1218–1223). IEEE.

Mammeri, A., Khiari, E.-H., & Boukerche, A. (2014). Road-sign text recognition architecture for intelligent transportation systems. In *2014 IEEE 80th vehicular technology conference (VTC Fall)* (pp. 1–5). IEEE.

Mishra, A., Alahari, K., & Jawahar, C. (2011). An MRF model for binarization of natural scene text. In *ICDAR-international conference on document analysis and recognition*. IEEE.

Mishra, A., Alahari, K., & Jawahar, C. (2012). Scene text recognition using higher order language priors. In *BMVC-British machine vision conference*. BMVA.

Neumann, L., & Matas, J. (2010). A method for text localization and recognition in real-world images. In *Asian conference on computer vision* (pp. 770–783). Springer.

Neumann, L., & Matas, J. (2012). Real-time scene text localization and recognition. In *2012 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3538–3545). IEEE.

Neumann, L., & Matas, J. (2013). On combining multiple segmentations in scene text recognition. In *2013 12th international conference on document analysis and recognition (ICDAR)* (pp. 523–527). IEEE.

Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., & Shiose, T. (2005). A novel adaptive morphological approach for degraded character image segmentation. *Pattern Recognition*, *38*(11), 1961–1975.

Parkinson, C., Jacobsen, J. J., Ferguson, D. B., & Pombo, S. A. (2016). Instant translation system, Nov. 29. US Patent 9,507,772.

Qin, S., Bissacco, A., Raptis, M., Fujii, Y., & Xiao, Y. (2019). Towards unconstrained end-to-end text spotting. In *Proceedings of the IEEE international conference on computer vision* (pp. 4704–4714).

Qiu, W., Zhong, F., Zhang, Y., Qiao, S., Xiao, Z., Kim, T. S., & Wang, Y. (2017). Unrealcv: Virtual worlds for computer vision. In *Proceedings of the 25th ACM international conference on multimedia* (pp. 1221–1224). ACM.

Phan, T. Q., Shivakumara, P., Tian, S., & Tan, C. L. (2013). Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 569–576).

Redmon, J., & Farhadi, A. (2017). Yolo9000: Better, faster, stronger. arXiv preprint.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 779–788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Rodriguez-Serrano, J. A., Gordo, A., & Perronnin, F. (2015). Label embedding: A frugal baseline for text recognition. *International Journal of Computer Vision*, *113*(3), 193–207.

Rodriguez-Serrano, J. A., Perronnin, F., & Meylan, F. (2013). Label embedding for text recognition. In *Proceedings of the British machine vision conference*. Citeseer.

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional networks for biomedical image segmentation*. Berlin: Springer.

Roy, P. P., Pal, U., Llados, J., & Delalandre, M. (2009). Multi-oriented and multi-sized touching character segmentation using dynamic programming. In *10th international conference on document analysis and recognition, 2009*. IEEE.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.

Schroth, G., Hilsenbeck, S., Huitl, R., Schweiger, F., & Steinbach, E. (2011). Exploiting text-related features for content-based image retrieval. In *2011 IEEE international symposium on multimedia* (pp. 77–84). IEEE.

Schulz, R., Talbot, B., Lam, O., Dayoub, F., Corke, P., Upcroft, B., & Wyeth, G. (2015). Robot navigation using human cues: A robot navigation system for symbolic goal-directed exploration. In *Proceedings of the 2015 IEEE international conference on robotics and automation (ICRA 2015)* (pp. 1100–1105). IEEE.

Sheshadri, K., & Divvala, S. K. (2012). Exemplar driven character recognition in the wild. In *BMVC* (pp. 1–10).

Shi, B., Bai, X., & Belongie, S. (2017a). Detecting oriented text in natural images by linking segments. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

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

Shi, B., Wang, X., Lyu, P., Yao, C., & Bai, X. (2016). Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4168–4176).

Shi, B., Yang, M., Wang, X., Lyu, P., Bai, X., & Yao, C. (2018). Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(11), 855–868.

Shi, C., Wang, C., Xiao, B., Zhang, Y., Gao, S., & Zhang, Z. (2013). Scene text recognition using part-based tree-structured character detection. In *2013 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2961–2968). IEEE.

Shivakumara, P., Bhowmick, S., Su, B., Tan, C. L., & Pal, U. (2011). A new gradient based character segmentation method for video text recognition. In *2011 international conference on document analysis and recognition (ICDAR)*. IEEE.

Su, B., & Lu, S. (2014). Accurate scene text recognition based on recurrent neural network. In *Asian conference on computer vision* (pp. 35–48). Springer.

Sun, Y., Liu, J., Liu, W., Han, J., Ding, E., & Liu, J. (2019). Chinese street view text: Large-scale Chinese text reading with partially supervised learning. In *Proceedings of the IEEE international conference on computer vision* (pp. 9086–9095).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Tian, S., Pan, Y., Huang, C., Lu, S., Yu, K., & Tan, C. L. (2015). Text flow: A unified text detection system in natural scene images. In *Proceedings of the IEEE international conference on computer vision* (pp. 4651–4659).

Tian, S., Lu, S., & Li, C. (2017). Wetext: Scene text detection under weak supervision. In *Proceedings of ICCV*.

Tian, Z. Huang, W., He, T., He, P., & Qiao, Y. (2016). Detecting text in natural image with connectionist text proposal network. In *In Proceedings of European conference on computer vision (ECCV)* (pp. 56–72). Springer.

Tian, Z., Shu, M., Lyu, P., Li, R., Zhou, C., Shen, X., & Jia, J. (2019). Learning shape-aware embedding for scene text detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4234–4243).

Tsai, S. S., Chen, H., Chen, D., Schroth, G., Grzeszczuk, R., & Girod, B. (2011). Mobile visual search on printed documents using text and low bit-rate features. In *18th IEEE international conference on image processing (ICIP)* (pp. 2601–2604). IEEE.

Tu, Z., Ma, Y., Liu, W., Bai, X., & Yao, C. (2012). Detecting texts of arbitrary orientations in natural images. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 1083–1090). IEEE.

Uchida, S. (2014). Text localization and recognition in images and video. In *Handbook of document image processing and recognition* (pp. 843–883). Springer.

Wachenfeld, S., Klein, H.-U., & Jiang, X. (2006). Recognition of screen-rendered text. In *18th international conference on pattern recognition, 2006. ICPR 2006* (Vol. 2, pp. 1086–1089). IEEE.

Wakahara, T., & Kita, K. (2011). Binarization of color character strings in scene images using k-means clustering and support vector machines. In *2011 international conference on document analysis and recognition (ICDAR)* (pp. 274–278). IEEE.

Wang, C., Yin, F., & Liu, C.-L. (2017). Scene text detection with novel superpixel based character candidate extraction. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)* (Vol. 1, pp. 929–934). IEEE.

Wang, F., Zhao, L., Li, X., Wang, X., & Tao, D. (2018). Geometry-aware scene text detection with instance transformation network.

In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1381–1389).
Wang, K., Babenko, B., & Belongie, S. (2011). End-to-end scene text recognition. In *2011 IEEE international conference on computer vision (ICCV),* (pp. 1457–1464). IEEE.
Wang, T., Wu, D. J., Coates, A., & Ng, A. Y. (2012). End-to-end text recognition with convolutional neural networks. In *2012 21st international conference on pattern recognition (ICPR)* (pp. 3304–3308). IEEE.
Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., & Shao, S. (2019a). Shape robust text detection with progressive scale expansion network. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.
Wang, X., Jiang, Y., Luo, Z., Liu, C.-L., Choi, H., & Kim, S. (2019b). Arbitrary shape scene text detection with adaptive text region representation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6449–6458).
Weinman, J., Learned-Miller, E., & Hanson, A. (2007). Fast lexicon-based scene text recognition with sparse belief propagation. In *ICDAR* (pp. 979–983). IEEE.
Wolf, C., & Jolion, J.-M. (2006). Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal of Document Analysis and Recognition (IJDAR)*, *8*(4), 280–296.
Wu, L., Zhang, C., Liu, J., Han, J., Liu, J., Ding, E., & Bai, X. (2019). Editing text in the wild. In *Proceedings of the 27th ACM international conference on multimedia* (pp. 1500–1508).
Wu, Y., & Natarajan, P. (2017). Self-organized text detection with minimal post-processing via border learning. In *Proceedings of the IEEE conference on CVPR* (pp. 5000–5009).
Xia, Y., Tian, F., Wu, L., Lin, J., Qin, T., Yu, N., & Liu, T.-Y. (2017). Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in neural information processing systems* (pp. 1784–1794).
Xing, L., Tian, Z., Huang, W., & Scott, M. R. (2019). Convolutional character networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 9126–9136).
Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048–2057).
Xue, C., Lu, S., & Zhan, F. (2018). Accurate scene text detection through border semantics awareness and bootstrapping. In *In Proceedings of European conference on computer vision (ECCV)*.
Yang, M., Guan, Y., Liao, M., He, X., Bian, K., Bai, S., et al. (2019). Symmetry-constrained rectification network for scene text recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 9147–9156).
Yang, Q., Jin, H., Huang, J., & Lin, W. (2020). Swaptext: Image based texts transfer in scenes. arXiv preprint arXiv:2003.08152.
Yang, X., He, D., Zhou, Z., Kifer, D., & Giles, C. L. (2017). Learning to read irregular text with attention mechanisms. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI-17* (pp. 3280–3286).
Yao, C., Bai, X., Shi, B., & Liu, W. (2014). Strokelets: A learned multi-scale representation for scene text recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4042–4049).
Yao, C., Bai, X., Sang, N., Zhou, X., Zhou, S., & Cao, Z. (2016). Scene text detection via holistic, multi-channel prediction. arXiv preprint arXiv:1606.09002.
Ye, Q., & Doermann, D. (2015). Text detection and recognition in imagery: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(7), 1480–1500.
Ye, Q., Gao, W., Wang, W., & Zeng, W. (2003). A robust text detection algorithm in images and video frames. In *IEEE ICICS-PCM* (pp. 802–806).
Yi, C., & Tian, Y. (2011). Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing*, *20*(9), 2594–2605.
Yin, F., Wu, Y.-C, Zhang, X.-Y., & Liu, C.-L. (2017). Scene text recognition with sliding convolutional character models. arXiv preprint arXiv:1709.01727.
Yin, X.-C., Yin, X., Huang, K., & Hao, H.-W. (2014). Robust text detection in natural scene images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(5), 970–983.
Yin, X.-C., Zuo, Z.-Y., Tian, S., & Liu, C.-L. (2016). Text detection, tracking and recognition in video: A comprehensive survey. *IEEE Transactions on Image Processing*, *25*(6), 2752–2773.
Yu, D., Li, X., Zhang, C., Han, J., Liu, J., & Ding, E. (2020). Towards accurate scene text recognition with semantic reasoning networks. arXiv preprint arXiv:2003.12294.
Yuan, T.-L., Zhu, Z., Xu, K., Li, C.-J., & Hu, S.-M. (2018). Chinese text in the wild. arXiv preprint arXiv:1803.00085.
Zhan, F., & Lu, S. (2019). ESIR: End-to-end scene text recognition via iterative image rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
Zhan, F., Lu, S., & Xue, C. (2018). Verisimilar image synthesis for accurate detection and recognition of texts in scenes.
Zhang, C., Liang, B., Huang, Z., En, M., Han, J., Ding, E., & Ding, X. (2019). Look more than once: An accurate detector for text of arbitrary shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.
Zhang, D., & Chang, S.-F. (2003). A Bayesian framework for fusing multiple word knowledge models in videotext recognition. In *Computer vision and pattern recognition, 2003.* IEEE.
Zhang, S., Liu, Y., Jin, L., & Luo, C. (2018). Feature enhancement network: A refined scene text detector. In *Proceedings of AAAI, 2018*.
Zhang, S.-X., Zhu, X., Hou, J.-B., Liu, C., Yang, C., Wang, H., & Yin, X.-C. (2020). Deep relational reasoning graph network for arbitrary shape text detection. arXiv preprint arXiv:2003.07493.
Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., & Bai, X. (2016). Multi-oriented text detection with fully convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.
Zhiwei, Z., Linlin, L., & Lim, T. C. (2010). Edge based binarization for video text images. In *2010 20th international conference on pattern recognition (ICPR)* (pp. 133–136). IEEE.
Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). EAST: An efficient and accurate scene text detector. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.
Zhu, Y., Yao, C., & Bai, X. (2016). Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, *10*(1), 19–36.
Zitnick, C. L., & Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *Proceedings of European conference on computer vision (ECCV)* (pp. 391–405). Springer.