Scene Text Detection and Recognition: The Deep Learning Era

Shangbang Long, Xin He, Cong Yao

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 inescapably influenced by this wave of revolution, consequentially entering the era of deep learning. In recent years, the community has witnessed substantial advancements in mindset, approach 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 the grand challenges still remained. We expect that this review paper would serve as a reference book for researchers in this field. Related resources are also collected and compiled in our Github repository: **https://github.com/Jyouhou/SceneTextPapers**.

Index Terms—Scene Text, Detection, Recognition, Deep Learning, Survey

1 Introduction

Text is one of the most brilliant creations of the humankind. It's the written form of human language, and bear cultural inheritance. On the one hand, extracting text from media such as documents and financial bills can save time and improve productivity in office and other application scenarios. On the other hand, text in image can provide extra information of the scene, and assist the understanding of it, which can be used in a wide range of applied Computer Vision (CV) tasks, such as *image-based search* [125], [144], e.g. for e-commerce and geolocation, *instant translation* [25], [111], *robots navigation* [23], [85], [86], [126], and *industrial automation* [18], [44], [52]. Therefore, scene text detection and recognition[1], as shown in Fig.1, have become a popular research topic.

Though we have seen rapid progresses as well as large-scale commercial deployment of this technology in the last several years, detecting and recognizing text from real world scene has been a non-trivial and challenging task ever since the machine learning era. Before deep learning rose to the main stream, researchers focused mainly on designing features by hand. This has been a brain-teasing task, as text are highly variant and complex in the following ways:

• **Complex Background** Scene text can appear in a variety of backgrounds, including but not limited to signs, walls, glasses and even hung in the air, which means that its background can be anything. Some backgrounds are noisy and disturbing in themselves, e.g. billboards that are glowing; glasses that you can see through; walls that have patterns or strips that look like text. Distinguishing text from its background is not a trivial task.

• *S. Long is with Peking University and work as an intern at Megvii (Face++), Beijing, China.*
E-mail: shangbang.long@pku.edu.cn
• *X. He and C. Yao are with Megvii (Face++), Beijing, China.*

Manuscript received July x, 2018; revised -, -.

1. In the industry, it has another more widely known name: Optical Character Recognition (OCR).

• **Varying Text** In contrast to document scanning, extracting text from natural scene is much more difficult as scene text are diverse. One characteristic is that, they have a diversity of shapes, colors, fonts, sizes, and orientations, while text in documents are usually clear and horizontally or vertically aligned, and have single color, size and font. In some conditions, the text are even decorated with varying patterns and LEDs.

• **Sensitivity and Interference** In general object detection, the shapes of the targets are unique to some extent. For example, it's unlikely that humans mistake a panda for an airplane. However, text have roughly the same shape, and only differ in details and minute patterns. Some characters share similar physical appearance. Environment noise can even obfuscate one character for another. Therefore, detection and recognition of text are sensitive to environment interference, e.g. lighting condition, blur, low resolution and partial occlusion.

• **Unique Characteristics of Text as a Special Object Type** Although the detection of text can be considered as a special case of object detection, it's distinguished by its unique complications. Text usually have varying aspect ratios, different orientations and even irregular shapes, i.e. curved text. Besides, since the recognition step depends on the quality of detected text region, the detection module is expected to extract text regions that are as tight and precise as possible. Varying aspect ratio is a challenge in itself. Tight prediction required by oriented and even curved text is also non-trivial.

These difficulties run through the years before deep learning showed its potential in CV as well as tasks in other fields. As deep learning came to prominence after AlexNet [72] won the ILSVRC2012 [124] contest, researchers can turn to deep learning models for automatic feature extraction and start with more in-depth researches. The community are now working on ever more challenging targets. The progresses made in recent years can be summarized as

Fig. 1: The concept of scene text detection and recognition. The image sample is from the Total-Text [17] dataset.

follows:

• **Incorporation of Deep Learning** Nearly all recent methods are built based on deep learning models. Most importantly, deep learning frees researchers from the exhausting work of 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. Besides, these algorithms provide significant improvements over previous ones. Gradient-based training routines also give rise to end-to-end trainable methods, further simplifying the traditional detector-recognizer split.

• **Target-Oriented Algorithms and Datasets** Researchers are now turning to more specific aspects and targets. Grounded in difficulties in real-scenario, newly published datasets are collected with unique and representative characteristics. For example, there are datasets that feature long text, blurred text, and curved text respectively. Driven by these datasets, almost all algorithms published in recent years are designed to tackle specific challenges. For example, some are proposed to detect oriented text, while others aim at blurred and unfocused scene images. These particular ideas are also combined to make more general purpose methods.

• **Advances in Auxiliary Technologies** Apart from new datasets and new models devoted to the main task, auxiliary technologies that do not solve the task directly also find their places in this field, e.g. synthetic datasets, bootstrapping, and etc..

In this survey, we present an overview of the recent development in the field of text detection and recognition, with focus on the deep learning era. We look back on these methods from different perspectives, and list the up-to-date datasets. We also analyze the status quo and predict future research trends.

There have already been several well-written and informative review papers [146], [165], [171], [184]. 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.

The remaining parts of this paper would be arranged as follows. In Section 2, we would briefly review the methods before the deep learning era. In Section 3, we talk about the development of deep learning techniques and introduce algorithms that are closely related to text detection and recognition. In Section 4, we list and summarize the algorithms based on deep learning in a hierarchical order. In Section 5, we take a look at the datasets and evaluation protocols. Finally, we list some newly developed applications and our opinions on the current status and future trends.

2 Methods before the Deep Learning Era

2.1 Overview

In this section, we take a brief glance retrospectively at text detection and recognition methods before the deep learning era. More detailed and comprehensive coverage of these works can be found in [146], [165], [171], [184]. For text detection and recognition, the attention has been the design of features. For end-to-end system, the design of pipeline is the main focus.

In this period of time, most text detection methods either adopt **Connected Components Analysis** (CCA) [26], [57], [63], [107], [145], [167], [170] or **Sliding Window** (SW) based classification [19], [75], [153], [155]. 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 [75], **Conditional Random Field** (CRF) [153] and other alternative graph based methods [19], [155].

For text recognition, one branch adopted the feature-based methods. Shi *et al.* [135] and Yao *et al.* [164] proposed **character segments** based recognition algorithms. Rodriguez *et al.* [118], [119] and Gordo *et al.* [40] and Almazan *et al.* [4] utilized **label embedding** to directly perform matching between strings and images. **Stoke** [12] and **character key-points** [113] are also detected as features for classification. Another discomposed the recognition process into a series of sub-problems. Various methods have been proposed to tackle these **sub-problems**, which includes text binarization [76], [102], [150], [179], text line segmentation [166], character segmentation [110], [123], [136], single character recognition [14], [129] and word correction [66], [103], [149], [156], [177].

There have been efforts devoted to integrated (i.e. end-to-end as we call it today) systems as well [106], [153]. In

Fig. 2: Illustration of traditional methods based on hand-crafted features: (1) Top: **Maximally Stable Extremal Regions** (MSER) based methods [107], assuming chromatic consistency within each character; (2) Bottom: **Stroke Width Transform** (SWT) based methods [26], assuming consistent stroke width within each character.

Wang *et al.* [153], characters are considered as a special case in object detection and detected by a nearest neighbor classifier trained on HOG features [21] and then grouped into words through a Pictorial Structure (PS) based model [28]. Neumann and Matas [106] 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 detected character segmentations using extremal regions and decoded 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 hand crafted image features, which entails demanding and repetitive pre-processing and post-processing steps. Constrained by the limited representation ability of hand crafted features and the complexity of pipelines, those methods can hardly handle intricate circumstances, e.g. blurred images in the ICDAR2015 dataset [67].

3 Development of Deep Learning

Recent years have witnessed the rapid rise of deep learning [38], which ultimately revolutionised the AI industry, including text detection and recognition. Deep learning is a set of learning algorithms that approximate a given mapping function by automatically learning to extract features from raw inputs and fit the output labels. A deep learning model usually consists a sequence of computation steps that are fully differentiable, so that the whole model can be optimized end-to-end with gradient descent methods applied to a proper training target.

Deep learning is applied in many fields in artificial intelligence, and has shown significant improvements over traditional machine learning methods. In this section, we briefly introduce the tasks and models that are closely related and fundamental to text detection and recognition.

3.1 Image Classification Task

Given an image and a set of candidate categories, an *Image Classification* model predicts the correct category that the image belongs to, e.g. *dog*, *car*, and etc.. Algorithms based on deep learning have gradually surpassed traditional methods and finally achieved better performance than human do [48], [56], [72], [138]. AlexNet [72] was the winner of the ImageNet Large Scale Visual Recognition Challenge [124] in 2012, achieving 10.8% less top-5 error rate than the runner-up. It uses a sequence of *Convolutional Neural Networks* (CNN), followed by several *Fully-Connected* (FC) layers, and predicts a probability distribution over all candidate categories. The filter size of lower-level CNNs is larger, while those in higher-level layers are smaller. Similarly, VGG [138] is composed of a sequence of CNN layers, but only 3×3 filter sizes except the first layer are used. It stacks a total number of 16 or 19 layers to increase the CNN's receptive field. To train deeper neural network, the 151-layered ResNet [48] was proposed. A residual connection (identity mapping in practice) from the input is added to the output for each CNN block (several layers of CNN). ResNet is the first algorithm that surpasses human performance.

Progresses in Image Classification have laid solid foundations for other CV tasks, as models in these tasks usually take advantage of off-the-shelf models from Image Classification works, which are termed as *base-net*, *backbone network*, or *stem-network*. The Image Classification task demonstrates the possibility of performing end-to-end learning, which can be shared among various tasks.

3.2 Object Detection Task

Object detection aims to detect, i.e. localize and recognize, objects of a given set of classes from an input image. There are mainly two branches, i.e. region-proposal based methods [34], [35], [116] and anchor-based methods [82], [114]. Both branches have indirectly inspired text detection algorithms based on them [78], [97].

3.2.1 *Region-proposal based*

The Region-based CNN (R-CNN) approach extracts a manageable number of candidate regions, and uses image classification model to predict whether it's a semantic object as well as its category. Fast-RCNN [34] accelerates the pipeline by applying region proposal to the extracted feature maps instead of the original images, in order to avoid repetitive computation. Faster-RCNN [116] consists of two stages. In the first stage, a Region Proposal Network (RPN) proposes candidate object bounding boxes. The second stage is similar to that of Fast-RCNN.

3.2.2 Anchor based

Anchor-based methods give predictions in one pass. Single Shot Detector (SSD) [82] and You-Only-Look-Once (YOLO) [114] uses similar structures. After passing the image into a sequence of CNNs, the final output is a feature map, where each position (x, y) is a feature vector representing the corresponding region in the input image. A classifier and position-regression are applied to the feature vector, predicting whether there is a semantic object, its class, and its precise position.

3.3 Semantic Segmentation Task

Semantic Segmentation is a task similar to object detection, where we need to predict the semantic category of *each pixel* in the original image instead. Fully Convolutional Network (FCN) [91] is proposed for this task. It consists of a sequence of CNN and pooling layers alternatively, followed by deconvolutional layers [175] so that the size of output layer is the same as the input image. The output layer is a feature map. The feature vector at each position is fed into a classifier that predicts the category of the pixel. Deconvolutional layer, in essence, is a modified CNN where the feed-forward and back-propagation are swapped. Therefore, the output feature map can be larger than input feature map, namely *up-sampling*. Later works introduce a pyramid-structure architecture [80], [101], [122], where feature maps from the *down-sampling* parts are added to the up-sampling side, to restore lower-level features. The incorporation of pyramid-structure connections is very important, as accurate pixel-level prediction would require more local features. Such techniques have been widely deployed in text detection and recognition models, e.g. EAST [180].

3.4 Sequence Modeling

Sequence modeling is an important task in natural language processing. In text detection and recognition, as the targets are themselves sequences of characters, it's necessary to consider sequence modeling methods. While previous methods use CRF or rule-based matching, deep learning based methods including sequence-to-sequence (Seq2Seq) learning [140] and attention-based Seq2Seq [6] are proposed and have achieved considerable improvements in the task of machine translation. Seq2Seq uses an encoder-decoder structure. The input sequence is first transformed into a sequence of word vector by word embedding method [100]. The encoder is an LSTM that reads the input sequence. The last hidden state of the encoder is used to initialized the decoder, which is also an LSTM. The decoder generates output sequence until it hits a stop symbol. We also refer readers to the following papers for recent advances in machine translation: Transformer [148], Convolutional Seq2Seq [31], [32], and the architecture evaluation survey [11]. These sequence modeling modules allow end-to-end gradient-ased learning for text recognition algorithms.

3.5 Initial Attempts in Text Detection and Recognition

Actually deep learning has already been used in a similar task decades ago: LeNet-5 [73] by LeCun *et al.* on the MNIST hand-written digit recognition task. It achieves an error rate of less than 1%, showing the ponderable potential of deep learning in CV tasks. The method of LeCun *et al.* credibly demonstrates the possibility for deep learning technology in CV tasks, where the input image is first represented as a 3-D array of real-valued numbers, and then passed to neural networks for subsequent tasks.

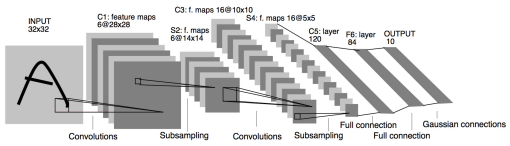

Fig. 3: Overview of LeNet-5, reprinted from [73].

4 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 proposals of new *methods*. Our reason for this 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 utilizes deep-learning based models; (2) Most researchers are approaching the problem from a diversity of perspectives. Methods driven by deep-learning enjoy the advantage that automatic feature learning can save us from designing and testing the large amount potential hand-crafted 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 [180], text of varying aspect ratios [130], and synthetic data [43]. 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 research by far. This is the most significant change compared to the former epoch.

In a nutshell, recent years have witness a blossoming expansion of research into subdivisible trends. We summarize these changes and trends in Fig.4, and we would follow this diagram in our survey.

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

4.1 Detection

There are three main trends in the field of text detection, and we would introduce them in the following sub-sections one by one. They are: (1) pipeline simplification; (2) changes in prediction units; (3) specified targets.

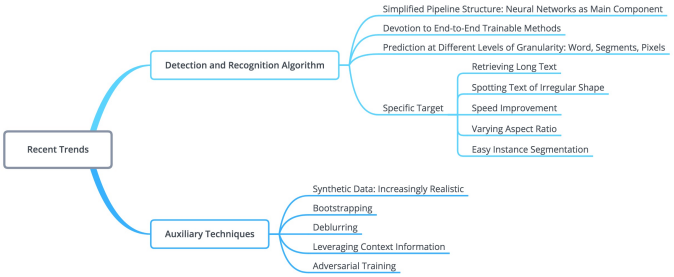

Fig. 4: Overview of recent progress and dominant trends.

4.1.1 Pipeline Simplification

One of the important trends are the simplification of the system pipeline. Most methods before the era of deep-learning, and some early methods that use deep-learning, have multi-step pipelines. More recent methods have simplified and much shorter pipelines, which is a key to reduce error propagation and simplify the training process. However, the main components of these methods are all end-to-end differentiable modules, i.e. deep learning models, which is an outstanding characteristic.

Multi-step methods: Early deep-learning based methods [163], [178][2], [46] cast the task of text detection into a multi-step process. In [163], a convolutional neural network is used to predict whether each pixel in the input image (1) belongs to a character, (2) is inside the text region, and (3) the text orientation around the pixel. As shown in Fig.x(a), connected positive responses are considered as a detection of character or text region. For characters belonging to the same text region, Delaunay triangulation [65] is applied, after which graph partition based on the predicted orientation attribute groups characters into text lines.

Similarly, [178] first predicts a dense map indicating which pixels are within text line regions. For each text line region, MSER [108] is applied to extract character candidates. Character candidates reveal information of the scale and orientation of the underlying text line. As the last step, minimum bounding box is extracted as the final text line candidate.

In [46], the detection process also consists of several steps. First, text blocks are extracted. Then the model crops and only focuses on the extracted text block to extract text center line(TCL), which is defined to be 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.

Simplified pipeline: More recent methods [49][3], [64], [78][4], [88], [130][5], [174][6], [97][7], [120], [79][8], [128] follow a 2-step pipeline, consisting of an end-to-end trainable neural network model and a post-processing step that is usually much simpler than previous ones. These methods mainly draw inspiration from techniques in general object detection [29], [34], [35], [47], [82], [116], and benefit from thehighly integrated neural network modules that can predict text instances directly. There are mainly two branches: (1) *Anchor-based methods* [49], [78], [88], [130] that predict the existence of text and regress the location offset only at pre-defined grid points of the input image; (2) *Region proposal methods* [64], [79], [97], [120], [128], [174] that predict and regress on the basis of extracted image region.

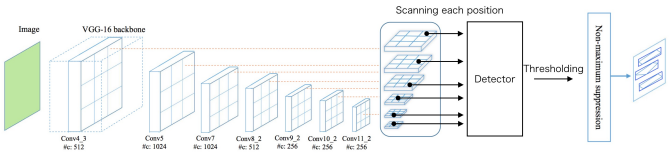

Fig. 5: Overview of TextBox, reprinted from [78].

Since the original targets of most of these works are not merely the simplification of pipeline, we only introduce some representative methods here. Other works will be introduced in the following parts.

Anchor-based methods draw inspiration from SSD [82], a general object detection network. As shown in Fig.5, a representative work, TextBoxes [78], adapts SSD network specially to fit the varying orientations and aspect-ratios of text line. Specifically, at each anchor point, default boxes are replaced by *default quadrilaterals*, which can capture the text line tighter and reduce noise.

A variant of the standard anchor-based default box prediction method is EAST [180][9]. In the standard SSD network, there are several feature maps of different sizes, on which default boxes of different receptive fields are detected. In EAST, all feature maps are integrated together by gradual upsampling, or U-Net [122] structure to be specific. The size of the final feature map is $\frac{1}{4}$ of the original input image, with c-channels. Under the assumption that each pixel only belongs to one text line, each pixel on the final feature map, i.e. the $1 \times 1 \times c$ feature tensor, is used to regress the rectangular or quadrilateral bounding box of the underlying text line. 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 the efficiency. Since EAST is most famous for its speed, we would re-introduce EAST in later parts, with emphasis on its efficiency.

Region proposal methods usually follow the standard object detection framework of r-cnn [34], [35], [116], where a simple and fast pre-processing method is applied, extracting a set of region proposals that could contain text lines. A neural network then classifies it as text/non-text and corrects the localization by regressing the boundary offsets. However, adaptations are necessary.

Rotation Region Proposal Networks [97] follows and adapts the standard Faster RCNN framework. In order to fit into text of arbitrary orientations, rotating region proposals

2. Code: https://github.com/stupidZZ/FCN_Text
3. Code: https://github.com/BestSonny/SSTD
4. Code: https://github.com/MhLiao/TextBoxes
5. Code: https://github.com/bgshih/seglink
6. Code: https://github.com/Yuliang-Liu/Curve-Text-Detector
7. Code: https://github.com/mjq11302010044/RRPN
8. Code: https://github.com/MhLiao/RRD
9. Code: https://github.com/zxytim/EAST

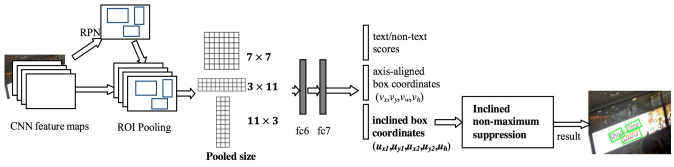

Fig. 6: Overview of R2CNN, reprinted from [64].

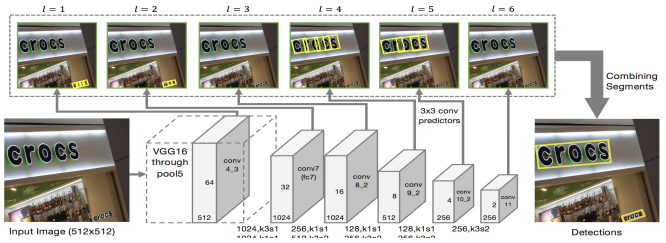

Fig. 7: Overview of SegLink, reprinted from [130].

are generated instead of the standard axis-aligned rectangles.

Similarly, R2CNN [64] makes modifications to the standard region proposal based object detection methods. As shown in Fig.6, to adapt to the varying aspects ratios, three Region of Interests Poolings of different sizes are performs, and concatenated for further prediction and regression. In FEN [128], adaptively weighted poolings are applied to integrated different pooling sizes. Textness scores are computed for poolings of 4 different sizes. The final prediction is made by leveraging the 4 scores.

4.1.2 Different Prediction Units

A main distinction between text detection and general object detection is that, text are homogeneous as a whole and show locality, while general object detection are not. By homogeneity and locality, we refer to the property that any part of a text instance is still text. Human do not have to see the whole text instance to find out the image is a text instance.

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.

In this part, we take the perspective of the granularity of text detection. There are two main level of prediction granularity, *text instance level* and *sub-text level*.

In **text instance level** methods [20], [51], [64], [78], [79], [88], [97], [128], [174], [180], detection of text follows the standard routine of general object detection, where a region-proposal network and a refinement network are combined to make predictions. The region-proposal network produces initial and coarse guess for the localization of possible text instance, and then a refinement part discriminates the proposals as text/non-text and also correct the localization of the text.

Contrarily, **sub-text level** detection methods [96], [22][10], [46], [159], [163], [49][11], [45], [130], [178], [143][12], [151], [183] only predicts parts that are combined to make a text instance. Such sub-text mainly includes *pixel*-level and *components*-level.

In **pixel-level** methods [22], [46], [49], [159], [163], [178], an end-to-end fully convolutional neural network learns to generate a dense prediction map indicting whether each pixel in the original image belongs to any text instances or not. Post-processing methods then groups pixels together depending on which pixels belong to the same text instance. 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 [22] learns to predict whether two adjacent pixels belong to the same text instance by adding link prediction to each pixel. Border learning method [159] casts each pixels into three categories: text, border, and background, assuming that border can well separate text instances. In Holistic [163], pixel-prediction maps include both text-block level and character center levels. Since the centers of characters do not overlap, the separation is done easily.

Since in this part we only intend to introduce the concept of prediction units, we would go back to details regarding the separation of text instances in the section of *Specific Targets*.

Components-level methods [45], [96], [130], [143], [151], [183] usually predicts at a medium granularity. Component refer to a local region of text instance, sometimes containing one or more characters.

As shown in Fig.7, SegLink [130] modified the original framework of SSD [82]. Instead of default boxes that represent whole objects, default boxes used in SegLink have only one aspect ratio and predict whether the covered region belongs to any text instances or not. The region is called *text segment*. Besides, links between default boxes are predicted, indicating whether the linked segments belong to the same text instance.

Corner localization methods [96] proposes to detect the 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.

SegLink [130] and Corner localization [96] are proposed specially for long and multi-oriented text. We only introduce the idea here and discuss more details in the section of *Specific Targets*, regarding how they are realized.

In a clustering based method [151], 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 instance.

Another branch of component-level method is Connectionist Text Proposal Network (CTPN) [143], [158], [183]. The CTPN models inherit the idea of anchoring and recurrent neural network for sequence labeling. These models usually consist of a CNN-based image classification network, e.g. VGG, and stack an RNN on top of it. Each position in the final feature map represents features in the region specified by the corresponding anchor. By assuming that text appear horizontally, each row of features are fed into a RNN or LSTM and labeled as text/non-text. Geometries are also predicted.

10. Code: https://github.com/ZJULearning/pixel_link
11. Code: https://github.com/BestSonny/SSTD
12. Code: https://github.com/tianzhi0549/CTPN

4.1.3 Specific Targets

Another characteristic of current text detection system is that, most of them are designed for special purposes, attempting to approach unique difficulties in detecting scene text. We broadly classify them into the following aspects.

4.1.3.1 **Long Text**: Unlike general object detection, text usually come in varying aspect ratios. They have much larger height-width ratio, and thus general object detection framework would fail. Several methods have been proposed [64], [96], [130], specially designed to detect long text.

R^2CNN [64] gives an intuitive solution, where ROI pooling with different sizes are used. Following the framework of Faster R-CNN [116], three ROI-poolings with varying pooling sizes, 7×7, 3×11, and 11×3, are performed for each box generated by region-proposal network, and the pooled features are concatenated for textness score.

Another branch learns to detect local sub-text components which are independent from the whole text [22], [96], [130]. SegLink [130] proposes to detect components, i.e. square areas that are text, and how these components are linked to each other. PixelLink [22] predicts which pixels belong to any text and whether adjacent pixels belong to the same text instances. Corner localization [96] detects text corners. All these methods learn to detect local components and then group them together to make final detections.

4.1.3.2 **Multi-Oriented Text**: Another distinction from general text detection is that text detection is rotation-sensitive and skewed text are common in real-world, while using traditional axis-aligned prediction boxes would incorporate noisy background that would affect the performance of the following text recognition module. Several methods have been proposed to adapt to it [64], [78], [79], [88], [97], [130], [180], [152][13].

Extending from general anchor-based methods, rotating default boxes [78], [88] are used, with predicted rotation offset. Similarly, rotating region proposals [97] are generated with 6 different orientations. Regression-based methods [64], [130], [180] predict the rotation and positions of vertexes, which are insensitive to orientation. Further, in Liao *et al.* [79], rotating filters [181] are incorporated to model orientation-invariance explicitly. The peripheral weights of 3×3 filters rotate around the center weight, to capture features that are sensitive to rotation.

While the aforementioned methods may entail additional post-processing, Wang *et al.* [152] proposes 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.

The core ideas behind these different methods are summarized in Fig.8

4.1.3.3 **Text of Irregular Shapes**: Apart from varying aspect ratios, another distinction is that text can have a diversity of shapes, e.g. curved text. Curved text poses a new challenge, since regular rectangular bounding box would incorporate a large proportion of background and even other text instances, making it difficult for recognition.

13. Code: https://github.com/zlmzju/itn

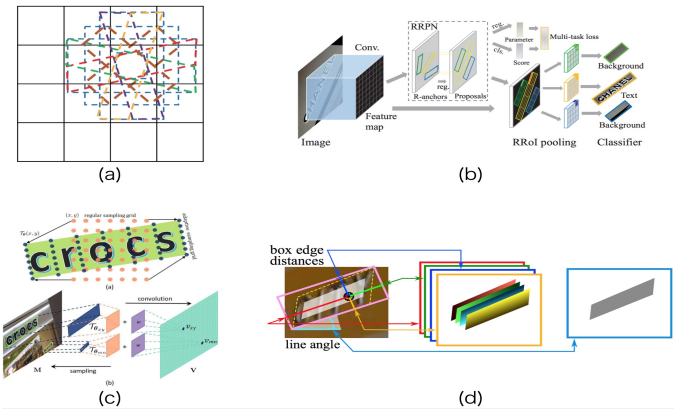

Fig. 8: Overview of different methods proposed for detecting multi-oriented text: (a) Rotating bounding boxes [88]; (b) Rotating regions of interest [97]; (c) Parametrized affine transformation layer [152]. Images are obtained from the original papers; (d) Direct regression of size and orientation [180].

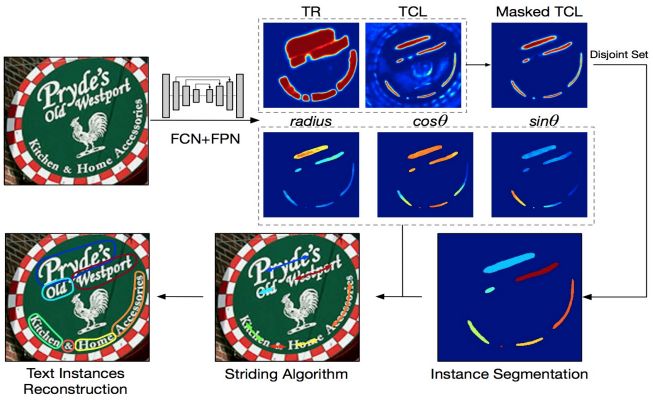

Fig. 9: Overview of TextSnake, reprinted from [92].

Extending from quadrilateral bounding box, it's natural to use bounding 'boxes' with more that 4 vertexes. Bounding polygons [174] with as many as 14 vertexes are proposed, followed by a bi-lstm [53] layer to refine the coordinates of the predicted vertexes. In their framework, however, axis-aligned rectangles are extracted as intermediate results in the first step, and the location bounding polygons are predicted upon them.

Similarly, Lyu *et al.* [95] modifies the Mask R-CNN [47] framework, so that for each region of interest—in the form of axis-aligned rectangles—character masks are predicted solely for each type of alphabets. These predicted characters are then aligned together to form a polygon as the detection results. Notably, they propose their method as an end-to-end system. We would refer to it again in the following part.

Viewing the problem from a different perspective, Long *et al.* [92] argues that text can be represented as a series of sliding round disks along the text center line (TCL), which accord with the running direction of the text instance. With the novel representation, they present a new model, *TextSnake*, as shown in Fig.9, that learns to predict local attributes, including TCL/non-TCL, text-region/non-text-region, radius, and orientation. The intersection of TCL

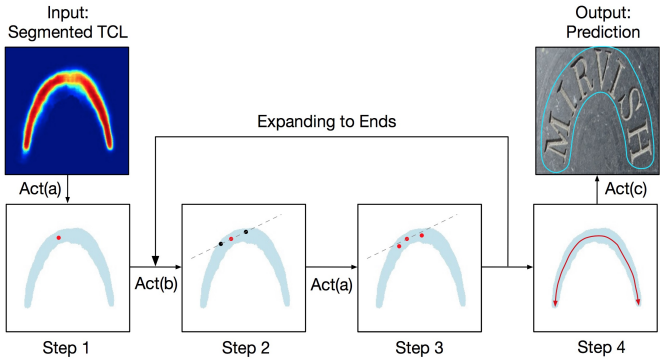

Fig. 10: Overview of the post-processing of TextSnake, reprinted from [92].

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 ordered point list, as demonstrated in Fig.10. With TCL and radius, the text line is reconstructed. It achieves state-of-the-art performance on several curved text dataset as well as more widely used ones, e.g. ICDAR2015 [67] and MSRA-TD500 [145]. Notably, Long *et al.* proposes 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.

4.1.3.4 **Speedup**: Current text detection methods place more emphasis on speed and efficiency, which is necessary for application in mobile devices.

The first work to gain significant speedup is EAST [180], which makes several modifications to previous framework. Instead of VGG [138], EAST uses PVANet [71] as its base-network, which strikes a good balance between efficiency and accuracy in the ImageNet competition. Besides, it simplifies the whole pipeline into a prediction network and a non-maximum suppression step. The prediction network is a U-shaped [122] fully convolutional network that maps an input image $I \in R^{H,W,C}$ to a feature map $F \in R^{H/4,W/4,K}$, where each position $f = F_{i,j,:} \in R^{1,1,K}$ is the feature vector that describes the predicted text instance. That is, the location of the vertexes or edges, the orientation, and the offsets of the center, for the text instance corresponding to that feature position (i, j). Feature vectors that corresponds to the same text instance are merged with the non-maximum suppression. It achieves state-of-the-art speed with FPS of 16.8 as well as leading performance on most datasets..

4.1.3.5 **Easy Instance Segmentation**: As mentioned above, recent years have witnessed methods with dense predictions, i.e. pixel level predictions [22], [46], [112], [159]. These methods generate a prediction map classifying each pixel as text or non-text. However, as text may come near each other, pixels of different text instances may be adjacent in the prediction map. Therefore, separating pixels become important.

Pixel-level text center line is proposed [46], since the center lines are far from each other. In [46], a prediction map indicating text lines is predicted. These text lines can be easily separated as they are not adjacent. To produce prediction for text instance, a binary map of text center line of a text instance is attached to the original input image and fed into a classification network. A saliency mask is generated indicating the detected text. However, this method involves several steps. The text-line generation step and the final prediction step can not be trained end-to-end, and error propagates.

Another way to separate different text instances is to use the concept of border learning [112], [159], [160], where each pixel is classified into one of the three classes: text, non-text, and text border. The text border then separates text pixels that belong to different instances. Similarly, in the work of Xue *et al.* [160], text are considered to be enclosed by 4 segments, i.e. a pair of long-side borders (*abdomen* and *back*) and a pair of short-side borders (*head* and *tail*). The method of Xue *et al.* is also the first to use DenseNet [56] as their basenet, which provides a consistant $2-4\%$ performance boost in F1-score over that with ResNet [48] on all datasets that it's evaluated on.

Following the linking idea of SegLink, PixelLink [22] learns to link pixels belonging to the same text instance. Text pixels are classified into groups for different instances efficiently via disjoint set algorithm. Treating the task in the same way, Liu *et al.* [90] proposes a method for predicting the composition of adjacent pixels with Markov Clustering [147], instead of neural networks. The Markov Clustering algorithm is applied to the saliency map of the input image, which is generated by neural networks and indicates whether each pixel belongs to any text instances or not. Then, the clustering results give the segmented text instances.

4.1.3.6 **Retrieving Designated Text**: Different from the classical setting of scene text detection, sometimes we want to retrieve a certain text instance given the description. Rong *et al.* [121] a multi-encoder framework to retrieve text as designated. Specifically, text is retrieved as required by a natural language query. The multi-encoder framework includes a Dense Text Localization Network (DTLN) and a Context Reasoning Text Retrieval (CRTR). DTLN uses an LSTM to decode the features in a FCN network into a sequence of text instance. CRTR encodes the query and the features of scene text image to rank the candidate text regions generated by DTLN. As much as we are concerned, this is the first work that retrieves text according to a query.

4.1.3.7 **Against Complex Background**: Attention mechanism is introduced to silence the complex background [49]. The stem network is similar to that of the standard SSD framework predicting word boxes, except that it applies inception blocks on its cascading feature maps, obtaining what's called Aggregated Inception Feature (AIF). An additional text attention module is added, which is again based on inception blocks. The attention is applied on all AIF, reducing the noisy background.

4.2 Recognition

In this section, we introduce methods that tackle the text recognition problem. Input of these methods are cropped text instance images which contain one word or one text line.

In traditional text recognition methods [9], [136], the task is divided into 3 steps, including image pre-processing,

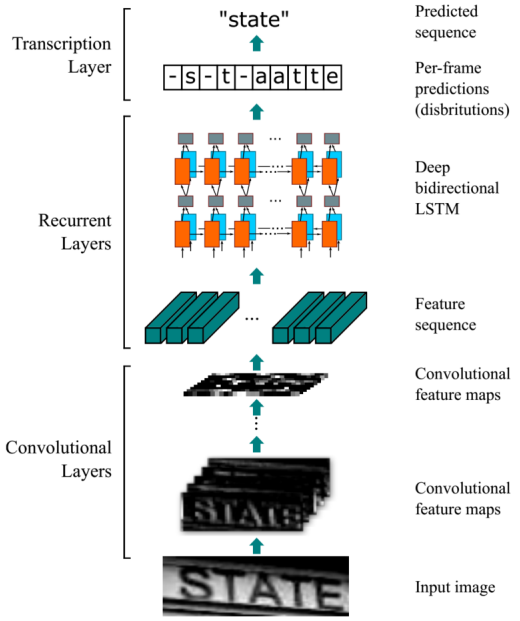

Fig. 11: The network architecture of CRNN [131].

character segmentation and character recognition. Character segmentation is considered the most challenging part due to the complex background and irregular arrangement of scene text, and largely constrained the performance of the whole recognition system. Two major techniques are adopted to avoid segmentation of characters, namely Connectionist Temporal Classification [41] and Attention mechanism. We introduce recognition methods in the literature based on which technique they employ, while other novel work will also be presented.

4.2.1 CTC-based Methods

CTC computes the conditional probability $P(L|Y)$, where $Y = y_1, ..., y_T$ represent the per-frame prediction of RNN and L is the label sequence, so that the network can be trained using only sequence level label as supervision. The first application of CTC in the OCR domain can be traced to the handwriting recognition system of Graves *et al.* [42]. Now this technique is widely adopted in scene text recognition [139], [84], [131][14], [30], [168].

Shi *et al.* [131] proposes a model that stacks CNN with RNN to recognize scene text images. As illustrated in Fig.11, CRNN consists of three parts: (1) convolutional layers, which extract a feature sequence from the input image; (2) recurrent layers, which predict a label distribution for each frame; (3) transcription layer (CTC layer), which translates the per-frame predictions into the final label sequence.

Instead of RNN, Gao *et al.* [30] 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. Overall difference with other frameworks are illustrated in Fig. 12

Yin *et al.* [168] also avoids using RNN in their model, they simultaneously detects and recognizes characters by

14. Code: https://github.com/bgshih/crnn

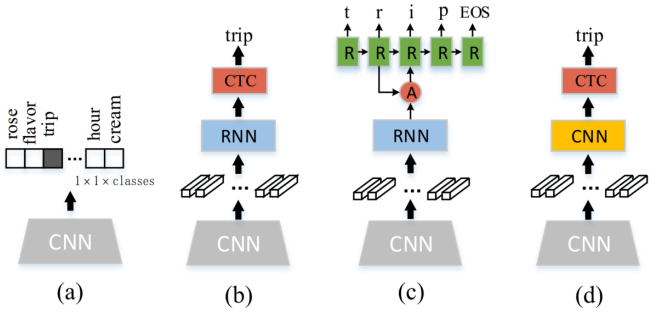

Fig. 12: Comparison of network architecture for scene text recognition. (a) CNN + softmax. (b) RNN + CTC. (c) RNN + Attention. (d) CNN + CTC. [30].

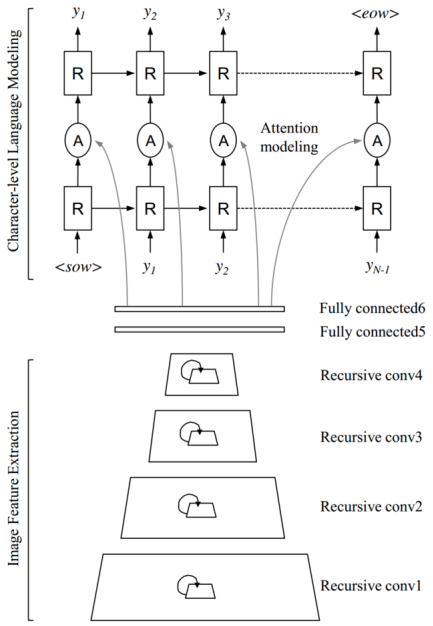

Fig. 13: The network architecture of R2AM [74].

sliding the text line image with character models, which are learned end-to-end on text line images labeled with text transcripts.

4.2.2 Attention-based methods

The attention mechanism was first presented in [6] to improve the performance of neural machine translation systems, and flourished in many machine learning application domains including Scene text recognition [15], [16], [33], [74], [89], [132], [161].

Lee *et al.* [74] presented a recursive recurrent neural networks with attention modeling (R2AM) for lexicon-free scene text recognition. the model first passes input images through recursive convolutional layers to extract encoded image features I, and then decodes them to output characters by recurrent neural networks with implicitly learned character-level language statistics. Attention-based mechanism performs soft feature selection for better image feature usage. The network architecture is depicted in Fig.13

Cheng *et al.* [15] observed the attention drift problem in existing attention-based methods and proposed an Focus

Attention Network (FAN) to attenuate it. The main idea is to add localization supervision to the attention module, while the alignment between image features and target label sequence are usually automatically learned in previous work.

In [7], Bai *et al.* proposed an edit probability (EP) metric to handle the misalignment between the ground truth string and the attention's output sequence of 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.

In [89], Liu *et al.* proposed an efficient attention-based encoder-decoder model, in which the encoder part is trained under binary constraints. Their recognition system achieves state-of-the-art accuracy while consumes much less computation costs than aforementioned methods.

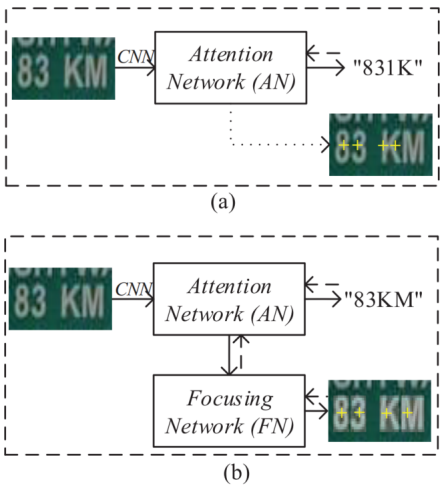

Fig. 14: Focusing network proposed in Cheng et al. [15] to tackle the attention drift problem.

Among those attention-based methods, some work made efforts to accurately recognize irregular (perspectively distorted or curved) text. Shi *et al.* [132], [133] proposed a text recognition system which combined a Spatial Transformer Network (STN) [61] and an attention-based Sequence Recognition Network. The STN predict a Thin-Plate-Spline transformations which rectify the input irregular text image into a more canonical form.

Yang *et al.* [161] introduced an auxiliary dense character detection task to encourage the learning of visual representations that are favorable to the text patterns.And they adopted 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.

In [16], Cheng *et al.* 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. And a weighting mechanism is designed to combine the four feature sequences.

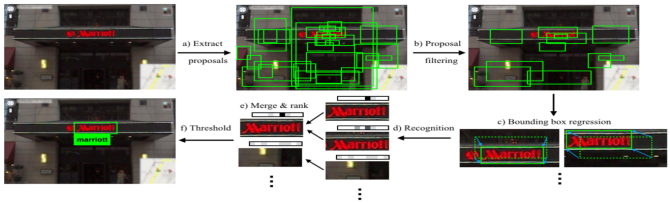

Fig. 15: The end-to-end text spotting pipeline introduced in [60].

Liu *et al.* [83] presented 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 an improved efficiency, and can handle different types of distortion that are hard to be modeled by a single global transformation.

4.2.3 Other Efforts

Jaderberg *et al.* [58], [59] performs word recognition on the whole image holistically. They train a deep classification neural network solely on data produced by a synthetic text generation engine, and achieve state-of-the-art performance on some benchmarks containing English words only. But application of this method is quiet limited since it cannot be applied to recognize long sequences such as phone numbers.

4.3 End-to-End System

In the past, text detection and recognition are usually cast as two independent sub-problems that are combined together to perform text retrieval 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. Efforts to build such systems have gained considerable momentum as a new trend.

While earlier work [153], [155] 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 [43], [60], [78]. Jaderberg *et al.* [60] use a combination of Edge Box proposals [185] and a trained aggregate channel features detector [24] to generate candidate word bounding boxes. Proposal boxes are filtered and rectified before being sent into their recognition model proposed in [59]. In [78], Liao *et al.* combined an SSD [82] based text detector and CRNN [131] to spot text in images. Lyu *et al.* [95] proposes a modification of Mask R-CNN that is adapted to produce shape-free recognition of scene text, as shown in Fig.17. For each region of interest, character maps are produced, indicating the existence and location of a single character. A post-processing that links these character together gives the final results.

One major drawbacks of the two-step methods is that the propagation of error between the text detection models and the text recognition models will lead to less satisfactory performance. Recently, more end-to-end trainable networks

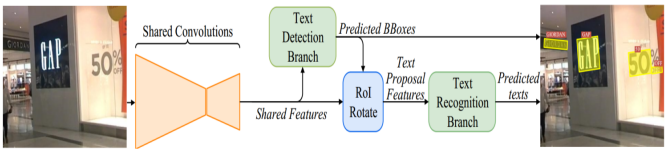

Fig. 16: End-to-End text spotting system in [87].

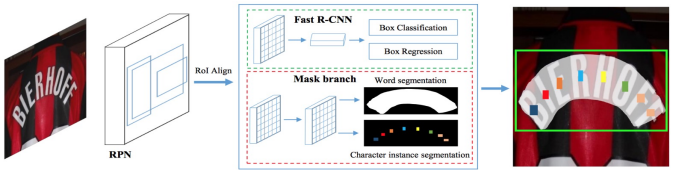

Fig. 17: Shape-free end-to-end text spotting system in [95].

are proposed to tackle the this problem [8][15], [13][16], [50], [77], [87].

Bartz *et al.* [8] presented an solution which utilize a STN [61] 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.* [77] substitute the object classification module in Faster-RCNN [116] with an encoder-decoder based text recognition model and make up their text spotting system. Lui *et al.* [87], Busta *et al.* [13] and He *et al.* [50] developed a unified text detection and recognition systems with a very similar overall architecture which consist of a detection branch and a recognition branch. Liu *et al.* [87] and Busta *et al.* [13] adopt EAST [180] and YOLOv2 [115] as their detection branch respectively, and have a similar text recognition branch in which text proposals are mapped into fixed height tensor by bilinear sampling and then transcribe in to strings by a CTC-based recognition module. He *et al.* [50] also adopted EAST [180] to generate text proposals, and they introduced character spatial information as explicit supervision in the attention-based recognition branch.

4.4 Auxiliary Techniques

Recent advances are not limited to detection and recognition models that aim to solve the task directly. We should also give credit to those auxiliary techniques that have played an important role. In this part, we briefly introduce some of the promising trends: synthetic data, bootstrapping, text deblurring, incorporating context information, and adversarial training.

4.4.1 Synthetic Data

Most deep learning models are data-thirsty. Their performance is guaranteed only when enough data are available. Therefore, artificial data generation has been a popular research topic, e.g. Generative Adversarial Nets (GAN) [39]. 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 [43], [59], [176] that can generate data instances of relatively high quality, and they have been widely used for pre-training models for better performance.

Jaderberg et at. [59] first proposes synthetic data for text recognition. Their method blends text with randomly cropped natural image 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 [43][17] first proposes to embed text in natural scene images for 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 [81] and semantic segmentation [5]. Semantic segmentation groups pixels together as semantic clusters, and each text instance is printed on one semantic surface, not overlapping multiple ones. Dense depth map is further used to determine the orientation and distortion of the text instance. Model trained only on SynthText achieves state-of-the-art on many text detection datasets. It's later used in other works [130], [180] as well for initial pre-training.

Further, Zhan *et al.* [176][18] equips 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.

4.4.2 Bootstrapping

Bootstrapping, or Weakly and semi supervision, is also important in text detection and recognition [55], [120], [142]. It's mainly used in word [120] or character [55], [142] level annotations.

Bootstrapping for word-box Rong *et al.* [120] proposes to combine an FCN-based text detection network with Maximally Stable Extremal Region (MSER) features to generate new training instances annotated on box-level. First, they train an FCN, which predicts the probability of each pixel belonging to text. Then, MSER features are extracted from regions where the text confidence is high. Using single linkage criterion (SLC) based algorithms [36], [137], final prediction is made.

Bootstrapping for character-box Character level annotations are more accurate and better. However, most existing datasets do not provide character-level annotating. Since character is smaller and close to each other, character-level annotation is more costly and inconvenient. There have been some work on semi-supervised character detection [55], [142]. The basic idea is to initialize a character-detector, and applies rules or threshold to pick the most reliable

15. Code: https://github.com/Bartzi/see
16. Code: https://github.com/MichalBusta/DeepTextSpotter
17. Code: https://github.com/ankush-me/SynthText
18. Code: https://github.com/fnzhan/Verisimilar-Image-Synthesis-for-Accurate-Detection-and-Recognition-of-Texts-in-Sc

predicted candidates. These reliable candidates are then used as additional supervision source to refine the character-detector. Both of them aim to augment existing datasets with character level annotations. They only differ in details.

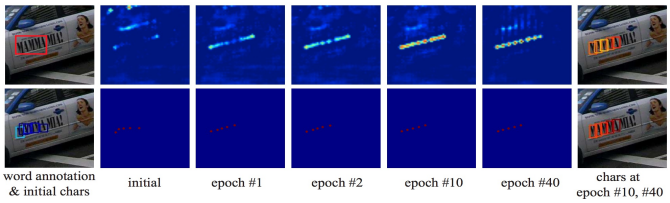

Fig. 18: Overview of the training process of WordSup, reprinted from [55].

WordSup [55] first initializes the character detector by training $5K$ warm-up iterations on synthetic dataset, as shown in Fig.18. 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:

$$\begin{aligned} s &= w \cdot s_1 + (1-w) \cdot s_2 \\ &= w \cdot \frac{area(B_{chars})}{area(B_{word})} + (1-w) \cdot (1 - \frac{\lambda_2}{\lambda_1}) \end{aligned} \quad (1)$$

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 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 a main characteristic for word instances in most datasets.

WeText [142] starts with a small datasets annotated on 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 is added to the initial small datasets and re-train the model.

4.4.3 Text Deblurring

By nature, text detection and recognition are more sensitive to blurring than general object detection. Some methods [54][19], [70] have been proposed for text deblurring.

Hradis *et al.* [54] proposes an FCN-based deblurring method. The core FCN maps the input image which is blurred and generates a deblurred image. They collect a dataset of well-taken images of documents, and process them with kernels designed to mimic hand-shake and defocus.

Khare *et al.* [70] proposes a quite different framework. Given a blurred image, g, it aims to alternatively optimize the original image f and kernel k by minimizing the following energy value:

$$E = \int (k(x,y) * f(x,y) - g(x,y))^2 dxdy + \lambda \int wR(k(x,y))dxdy$$

where λ is the regularization weight, with operator R as the Gaussian weighted (w) $L1$ norm. The optimization is done by alternatively optimizing over the kernel k and the original image f.

4.4.4 Context Information

Another way to make more accurate predictions is to take into account the context information. Intuitively, we know that text only appear on a certain surfaces, e.g. billboards, books, and etc.. Text are less likely to appear on the face of a human or an animal. Following this idea, Zhu *et al.* [182] proposes to incorporate the semantic segmentation result as part of the input. The additional feature filters out false positives where the patterns look like text.

4.4.5 Adversarial Attack

Text detection and recognition has a broad range of application. In some scenarios, the security of the applied algorithms becomes a key factor, e.g. autonomous vehicles and identity verification. Yuan *et al.* [173] proposes the first adversarial attack algorithm for text recognition. They propose a white-box attack algorithm that induces a trained model to generate a desired wrong output. Specifically, they aim to optimize a joint target of: (1) $D(x, x')$ for minimizing the alteration applied to the original image; (2) $L(x_{targeted})$ for the loss function with regard to the probability of the targeted output. They adapt the automated weighting method proposed by Kendall *et al.* [69] to find the optimum weight of the two targets. Their method realizes a success rate over 99.9% with $3-6\times$ speedup compared to other state-of-the-art attack methods. Most importantly, their method showed a way to carry out sequential attack.

5 Benchmark Datasets and Evaluation Protocols

As cutting edge algorithms achieved better on previous datasets, researchers were able to tackle more challenging aspects of the problems. New datasetes 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 performance on the widely used datasets when applicable.

5.1 Benchmark Datasets

We collect existing datasets and summarize their features in Tab.1. We also select some representative image samples from some of the datasets, which are demonstrated in Fig.19.

5.1.1 Datasets with both detection and recognition tasks

- The ICDAR 2003&2005 and 2011&2013

Held in 2003, the ICDAR 2003 Robust Reading Competition [94] is the first such benchmark dataset that's ever released for scene text detection and recognition[20]. Among

19. Code: http://www.fit.vutbr.cz/~ihradis/CNN-Deblur/

20. http://www.iapr-tc11.org/mediawiki/index.php/ICDAR_2003_Robust_Reading_Competitions

Fig. 19: Selected samples from ICDAR2013/2015/2017, Total-Text, CTW, CTW1500, and MSRA-TD500. Note that Total-Text provides two sets of annotations: rectangle (in *red*) and polygon (in *green*).

the 509 images, 258 are used for training and 251 for testing. The dataset is also used in ICDAR 2005 Text Locating Competition [93]. ICDAR 2015 also includes a digit recognition track[21].

In the ICDAR 2011 [22] and 2013 [23] Robust Reading Competitions, previous datasets are modified and extended, which make the new ICDAR 2011 [127] and 2013 [68] datasets. Problems in previous datasets are corrected, e.g. imprecise bounding boxes. State-of-the-art results are shown in Tab.2 for detection and Tab.**??** for recognition.

• ICDAR 2015 [24]

In real world application, images containing text may be too small, blurred, or occluded. To represent such a challenge, ICDAR 2015 is proposed as the Challenge 4 of the 2015 Robust Reading Competition [67] for incidental scene text detection. Scene text images in this dataset are taken by Google Glasses without taking care of the image quality. A large proportion of images are very small, blurred, and multi-oriented. There are 1000 images for training and 500 images for testing. The text instances from this dataset are labeled as word level quadrangles. State-of-the-art results are shown in Tab.3 for detection and Tab.**??** for recognition.

• ICDAR 2017 RCTW[25]

In ICDAR2017 Competition on Reading Chinese Text in the Wild [134], Shi *et al.* propose a new dataset, called CTW-12K, which mainly consists of Chinese. It is comprised of $12,263$ images in total, among which $8,034$ are for training and $4,229$ are for testing. Text instances are annotated with parallelograms. It's the first large scale Chinese dataset, and was also the largest published one by then.

• CTW [26]

The Chinese Text in the Wild (CTW) dataset proposed by Yuan *et al.* [172] is the largest annotated dataset to date. It has $32,285$ high resolution street view image of Chinese text, with $1,018,402$ character instances in total. All images are annotated at the character level, including its underlying character type, bouding box, and 6 other attributes. These attributes indicate whether its background is complex, whether it's raised, whether it's hand-written or printed, whether it's occluded, whether it's distorted, whether it uses word-art. The dataset is split into a training set of $25,887$ images with $812,872$ characters, a recognition test set of $3,269$ images with $103,519$ characters, and a detection test set of $3,129$ images with $102,011$ characters.

• Total-Text[27]

Unlike most previous datasets which only include text that are in straight lines, Total-Text consists of 1555 images with more than 3 different text orientations: Horizontal, Multi-Oriented, and Curved. Text instances in Total-Text are annotated with both quadrilateral boxes and polygon boxes of a variable number of vertexes. State-of-the-art results for Total-Text are shown in Tab.4 for detection and Tab.5 for recognition.

• SVT[28]

The Street View Text (SVT) dataset [153], [154] is a collection of street view images. SVT has 350 images. It only has word-level annotations.

• CUTE80 (CUTE) [29]

CUTE is proposed in [117]. The dataset focuses on curved text. It contains 80 high-resolution images taken in natural scenes. No lexicon is provided.

21. http://www.iapr-tc11.org/mediawiki/index.php?title=ICDAR_2005_Robust_Reading_Competitions
22. http://www.cvc.uab.es/icdar2011competition/
23. http://dagdata.cvc.uab.es/icdar2013competition/
24. http://rrc.cvc.uab.es/?ch=4&com=introduction
25. http://u-pat.org/ICDAR2017/program_competitions.php
26. https://ctwdataset.github.io
27. https://github.com/cs-chan/Total-Text-Dataset
28. http://www.iapr-tc11.org/mediawiki/index.php?title=The_Street_View_Text_Dataset
29. http://cs-chan.com/downloads_CUTE80_dataset.html

TABLE 1: Existing datasets: * indicates datasets that are the most widely used across recent publications. Newly published ones representing real-world challenges are marked **in bold**. En stands for English and Ch stands for Chinese.

Dataset (Year)	Image Num (train/test)	Text Num (train/test)	Orientation	Language	Characteristics	Detection Task	Recognition Task
ICDAR03 (2003)	509 (258/251)	2276 (1110/1156)	Horizontal	En	-	✓	✓
***ICDAR13 Scene Text(2013)**	462 (229/233)	- (848/1095)	Horizontal	En	-	✓	✓
***ICDAR15 Incidental Text(2015)**	1500 (1000/500)	- (-/-)	Multi-Oriented	En	*Blur Small Defocused*	✓	✓
ICDAR17 RCTW(2017)	12263 (8034/4229)	- (-/-)	Multi-Oriented	Ch	-	✓	✓
Total-Text (2017)	1555 (1255/300)	- (-/-)	*Multi-Oriented Curved*	En, Ch	Irregular polygon label	✓	✓
SVT (2010)	350 (100/250)	904 (257/647)	Horizontal	En	-	✓	✓
***CUTE (2014)**	80 (-/80)	- (-/-)	Curved	En	-	✓	✓
CTW (2017)	*32K* (*25K/6K*)	*1M* (*812K/205K*)	Multi-Oriented	Ch	*Fine-grained annotation*	✓	✓
***MSRA-TD500 (2012)**	500 (300/200)	1719 (1068/651)	Multi-Oriented	En, Ch	*Long text*	✓	-
HUST-TR400 (2014)	400 (400/-)	- (-/-)	Multi-Oriented	En, Ch	*Long text*	✓	-
ICDAR17 RRC-MLT(2017)	18000 (9000/9000)	- (-/-)	Multi-Oriented	*9 langanges*	-	✓	-
CTW1500 (2017)	1500 (1000/500)	- (-/-)	*Multi-Oriented Curved*	En	Bounding box with 14 vertexes	✓	-
IIIT 5K-Word (2012)	5000 (-/-)	5000 (2000/3000)	Horizontal	-	-	-	✓
SVTP (-)	639 (-/-)	639 (-/-)	Multi-Oriented	En	*Perspective text*	-	✓
SVHN (2010)	- (-/-)	600000 (-/-)	Horizontal	-	House number digits	-	✓

5.1.2 Datasets with only detection task

• MSRA-TD500[30] and HUST-TR400 [31]

The MSRA Text Detection 500 Dataset (MSRA-TD500) [145] is a benchmark dataset featuring long and multi-oriented text. Text instances in MSRA-TD500 have much larger aspect ratios than other datasets. Later, an additional set of images, called HUST-TR400, are published. HUST-TR400 [162] are collected in the same way as MSRA-TD500, usually used as a supplement to the MSRA-TD500 training data.

• ICDAR2017 RRC-MLT[32]

The dataset of ICDAR2017 RRC-MLT Challenge [104] contains $18K$ images with scripts of 9 languages, $2K$ for each. It features the largest number of languages up till now.

• SCUT-CTW1500 (CTW1500)[33]

CTW1500 is another dataset which features curved text. It consists of 1000 training images and 500 test images.Annotations in CTW1500 are polygons with 14 vertexes. Performances on CTW1500 are shown in Tab.6 for detection.

5.1.3 Datasets with only recognition task

• IIIT 5K-Word[34]

IIIT 5K-Word [103] is the largest 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. There are 5000 images in total, 2000 for training and 3000 for testing.

30. http://www.iapr-tc11.org/mediawiki/index.php/MSRA_Text_Detection_500_Database_(MSRA-TD500)

31. http://mclab.eic.hust.edu.cn/UpLoadFiles/dataset/HUST-TR400.zip

32. http://rrc.cvc.uab.es/?ch=8

33. https://github.com/Yuliang-Liu/Curve-Text-Detector

34. http://cvit.iiit.ac.in/projects/SceneTextUnderstanding/IIIT5K.html

TABLE 2: State-of-the-art detection performance on ICDAR2013. * means multi-scale, † stands for the base net of the model is not VGG16. The performance is based on DetEval.

Method	Precision	Recall	F-measure	FPS
Zhang *et al.* [178]	88	78	83	-
SynthText [43]	92.0	75.5	83.0	-
Holistic [163]	88.88	80.22	84.33	-
PixelLink [22]	86.4	83.6	84.5	-
CTPN [143]	93	83	88	7.1
He *et al.** [46]	93	79	85	-
SegLink [130]	87.7	83.0	85.3	20.6
He *et al.* * † [51]	92	80	86	1.1
TextBox++ [78]	89	83	86	1.37
EAST [180]	92.64	82.67	87.37	-
SSTD [49]	89	86	88	7.69
Lyu *et al.* [96]	93.3	79.4	85.8	10.4
Liu *et al.* [90]	88.2	87.2	87.7	-
He *et al.** [50]	88	87	88	-
Xue *et al.*† [160]	91.5	87.1	89.2	-
WordSup * [55]	93.34	87.53	90.34	-
Lyu *et al.** [95]	94.1	88.1	91.0	4.6
FEN [128]	93.7	90.0	92.3	1.11

TABLE 3: State-of-the-art detection performance on ICDAR2015. * means multi-scale, † stands for the base net of the model is not VGG16.

Method	Precision	Recall	F-measure	FPS
Zhang *et al.* [178]	71	43.0	54	-
CTPN [143]	74	52	61	7.1
Holistic [163]	72.26	58.69	64.77	-
He *et al.** [46]	76	54	63	-
SegLink [130]	73.1	76.8	75.0	-
SSTD [49]	80	73	77	-
EAST [180]	83.57	73.47	78.20	13.2
He *et al.* * † [51]	82	80	81	-
R2CNN [64]	85.62	79.68	82.54	0.44
Liu *et al.* [90]	72	80	76	-
WordSup * [55]	79.33	77.03	78.16	-
Wang *et al.*† [152]	85.7	74.1	79.5	-
Lyu *et al.* [96]	**94.1**	70.7	80.7	3.6
TextSnake [92]	84.9	80.4	82.6	1.1
He *et al.** [50]	84	83	83	-
Lyu *et al.** [95]	85.8	81.2	83.4	4.8
PixelLink [22]	85.5	82.0	83.7	3.0

TABLE 4: State-of-the-art detection performance on Total-Text

Method	Precision	Recall	F-measure
DeconvNet [109]	33	40	36
Lyu *et al.** [95]	69.0	55.0	61.3
TextSnake [92]	82.7	74.5	78.4

TABLE 5: State-of-the-art End-to-End performance on Total-Text. *None* refers to recognition without any lexicon; *Full* lexicon contains all words in test set.

Method	None	Full
TextBoxes *et al.* [78]	36.3	48.9
Lyu *et al.** [95]	52.9	71.8

TABLE 6: State-of-the-art detection performance on CTW1500.

Method	Precision	Recall	F-measure
SegLink [130]	42.3	40.0	40.8
EAST [180]	78.7	49.1	60.4
DMPNet [88]	69.9	56.0	62.2
CTD [174]	74.3	65.2	69.5
CTD+TLOC [174]	77.4	69.8	73.4
TextSnake [92]	67.9	85.3	75.6

TABLE 7: State-of-the-art detection performance on MSRA-TD500. † stands for models whose base nets are not VGG16.

Method	Precision	Recall	F-measure	FPS
Kang *et al.* [65]	71	62	66	-
Zhang *et al.* [178]	83	67	74	-
Holistic [163]	76.51	75.31	75.91	-
He *et al.* † [51]	77	70	74	-
EAST † [180]	87.28	67.43	76.08	13.2
Wu *et al.* [159]	77	78	77	-
SegLink [130]	86	70	77	8.9
PixelLink [22]	83.0	73.2	77.8	
TextSnake [92]	83.2	73.9	78.3	1.1
Xue *et al.*† [160]	83.0	77.4	80.1	-
Wang *et al.*† [152]	90.3	72.3	80.3	-
Lyu *et al.* [96]	87.6	76.2	81.5	5.7
Liu *et al.* [90]	88	79	83	-

• SVT-Perspective (SVTP)

SVTP is proposed in [113] for evaluating the performance of recognizing perspective text. Images in SVTP are picked from the side-view images in Google Street View. Many of them are heavily distorted by the non-frontal view angle. The dataset consists of 639 cropped images for testing, each with a 50-word lexicon inherited from the SVT dataset.

• SVHN[35] The street view house numbers (SVHN) dataset [105] contains more than 600000 digits of house numbers in natural scenes. The images are collected from Google View images. This dataset is usually used in digit recognition.

TABLE 8: Characteristics of the three vocabulary lists used in ICDAR 2013/2015. *S* stands for *Strongly Contextualised*, *W* for *Weakly Contextualised*, and *G* for *Generic*

Vocab List	Description
S	a per-image list of 100 words all words in the image + seletected distractors following the setup of Wang *et al.* [153]
W	all words in the entire test set 3 characters or longer, only letters
G	any vocabulary a $90k$-word vocabulary is provided

5.2 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 a 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.

5.2.1 Evaluation Protocols for 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.

• PASCAL [27]: 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.

• 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.

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

5.2.1.1 ICDAR2003/2005: The match score m is calculated in a way similar to IOU. It's defined as the ratio of the area of intersection over that of the minimum bounding rectangular bounding box containing both. The precision and recall is calculated as the mean match scores for the predicted instances and ground truth ones respectively:

$$precision = \frac{\sum_{r_P} m(r_P; GT)}{|P|}$$

and

$$recall = \frac{\sum_{r_{GT}} m(r_{GT}; P)}{|GT|}$$

where P is the set of predicted text instances and GT is the set of ground truth ones.

5.2.1.2 ICDAR2011/13: One major drawback of the evaluation protocol of ICDAR2003/2005 is that it only considers one-to-one match. It does not consider one-to-many, many-to-many, and many-to-one matchings, which underestimates the actual performance. Therefore, ICDAR2011/2013 follows the method proposed by Wolf *et al.* [157]. Precision and recall are computed as follows:

35. http://www.iapr-tc11.org/mediawiki/index.php?title=The_Street_View_House_Numbers_(SVHN)_Dataset

$$precision(G, D, t_r, t_p) = \frac{\sum_j Match_D(D_j, G, t_r, t_p)}{|D|}$$

and

$$recall(G, D, t_r, t_p) = \frac{\sum_i Match_D(G_i, D, t_r, t_p)}{|G|}$$

where G and D are the ground truth set and detection set; t_r and t_p are thresholds value for area precision and recall respectively, set to 0.8 and 0.4 in practice.

The match score function, $Match_D$ and $Match_G$, gives different score for each types of matching:

$$Match_{D/G}(D_j, G, t_r, t_p) = \begin{cases} 1, & one-to-one\ match \\ 0, & if\ no\ match \\ f_{sc}(k), & if\ many\ matches \end{cases} \quad (2)$$

$f_{sc}(k)$ is a function for punishment of many-matches, controlling the amount of splitting or merging. In practice, it's set to a constant function of 0.8.

5.2.1.3 MSRA-TD500: Yao *et al.* [145] proposes a new evaluation protocol for rotated bounding box, where both the predicted and ground truth bounding box are revolved horizontal 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 are less a pre-defined value (in practice $\frac{pi}{4}$).

5.2.2 Evaluation Protocols for Text Recognition and End-to-End System

Text recognition is another task where a cropped image is given which contains exactly one text instance, and we need to extract the text content from the image in a form that a computer program can understand directly, e.g. *string* type in C++ or *str* type in Python. There is not need for matching in this task. 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 is 100% correct). ICDAR also introduces an edit-distance based performance evaluation. Note that in end-to-end evaluation, matching is first performed in a similar way to that of text detection. State-of-the-art recognition performance on the most widely used datasets are summarized in Tab.9

The evaluation for end-to-end system is a combination of both detection and recognition. Given output from the system to be evaluated, i.e. text location and recognized content, predicted text instances are first matched with ground truth instances, followed by comparison of the text content.

The most widely used datasets for end-to-end systems are ICDAR2013 [68] and ICDAR2015 [67]. The evaluation over these two datasets are carried out under two different settings [1], 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 appear in a predesignated vocabulary, while other text instances are ignored. On the contrary, all text instances that

TABLE 9: State-of-the-art recognition performance across a number of datasets. "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.

Methods	ConvNet, Data	IIIT5k			SVT		IC03			IC13	IC15	SVTP	CUTE
		50	1k	0	50	0	50	Full	0	0	0	0	0
Wang *et al.* [153]	-	-	-	-	57.0	-	76.0	62.0	-	-	-	-	-
Bissacco *et al.* [9]	-	-	-	-	-	-	90.4	78.0	-	87.6	-	-	-
Almazan *et al.* [4]	-	91.2	82.1	-	89.2	-	-	-	-	-	-	-	-
Yao *et al.* [164]	-	80.2	69.3	-	75.9	-	88.5	80.3	-	-	-	-	-
Rodríguez-Serrano *et al.* [118]	-	76.1	57.4	-	70.0	-	-	-	-	-	-	-	-
Jaderberg *et al.* [62]	-	-	-	-	86.1	-	96.2	91.5	-	-	-	-	-
Su and Lu [139]	-	-	-	-	83.0	-	92.0	82.0	-	-	-	-	-
Gordo [40]	-	93.3	86.6	-	91.8	-	-	-	-	-	-	-	-
Jaderberg *et al.* [60]	VGG, 90k	97.1	92.7	-	95.4	80.7	98.7	**98.6**	93.1	90.8	-	-	-
Jaderberg *et al.* [58]	VGG, 90k	95.5	89.6	-	93.2	71.7	97.8	97.0	89.6	81.8	-	-	-
Shi *et al.* [131]	VGG, 90k	97.8	95.0	81.2	97.5	82.7	98.7	98.0	91.9	89.6	-	-	-
*Shi *et al.* [132]	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 *et al.* [74]	VGG, 90k	96.8	94.4	78.4	96.3	80.7	97.9	97.0	88.7	90.0	-	-	-
Yang *et al.* [161]	VGG, Private	97.8	96.1	-	95.2	-	97.7	-	-	-	-	75.8	69.3
Cheng *et al.* [15]	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.* [133]	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**

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 Tab.8. Note that under *End-to-End*, these vocabulary can still serve as reference.

State-of-the-art performance of End-to-End and Word Spotting tasks on ICDAR2013 and ICDAR2015 are summarized in Tab.11 and Tab.10 respectively.

6 Application

The detection and recognition of text—the visual and physical carrier of human civilization—allows 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.

Document Digitization So far, paper-based documents are the main storage medium for data in a wide diversity of industries. These documents, across multiple languages and various formats, can be scanned and digitized into structured forms with proper OCR techniques that machines can read. Potential beneficiaries may include: medical records, banking records, wills, financial records, history materials, paper books and etc.. Despite the recent digitization of information, there are still a large amount of paper-based documents from dozens of years ago. Neither are all the data produced today created in digital form. Digitization via OCR can make storage easier (consider the space needed, fireproofing, pest-control, oxidation damage, and etc.), and also more accessible to users. It also allows massive data analysis on them. A famous example is the Project Gutenberg [2] that digitizes and makes archives for books.

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 handwritten 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 this technologies, e.g. SF-Express[36]. Another potential application is note taking, such as NEBO[37], a note-taking software on tablets like iPad that can perform instant transcription as user writes 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++)[38].

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 vehicle, 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 [98], [99]. The largest dataset so far, CTW [172], also places extra emphasis on traffic signs. Another example is instant translation, where

36. Official website: http://www.sf-express.com/cn/sc/
37. Official website: https://www.myscript.com/nebo/
38. Megvii's AI cloud platform: https://www.faceplusplus.com/face-based-identification/

TABLE 10: State-of-the-art performance of End-to-End and Word Spotting tasks on ICDAR2015. * means multi-scale, † stands for the base net of the model is not VGG16.

Method	Word Spotting			End-to-End			FPS
	S	W	G	S	W	G	
Baseline OpenCV3.0+Tesseract [67]	14.7	12.6	8.4	13.8	12.0	8.0	-
TextSpotter [78]	37.0	21.0	16.0	35.0	20.0	16.0	1
Stradvision [67]	45.9	-	-	43.7	-	-	-
Deep2Text-MO [60], [169], [170]	17.58	17.58	17.58	16.77	16.77	16.77	-
TextProposals+DictNet [37], [59]	56.0	52.3	49.7	53.3	49.6	47.2	0.2
HUST_MCLAB [130], [131]	70.6	-	-	67.9	-	-	-
Deep Text Spotter [13]	58.0	53.0	51.0	54.0	51.0	47.0	9.0
FOTS* [87]	87.01	82.39	67.97	83.55	79.11	65.33	3.7
He *et al.* [50]	85	80	65	82	77	63	-
Mask TextSpotter [95]	79.3	74.5	64.2	79.3	73.0	62.4	2.6

TABLE 11: State-of-the-art performance of End-to-End and Word Spotting tasks on ICDAR2013. * means multi-scale, † stands for the base net of the model is not VGG16.

Method	Word Spotting			End-to-End			FPS
	S	W	G	S	W	G	
Jaderberg *et al.* [60]	90.5	-	76	86.4	-	-	-
FCRNall+multi-filt [43]	-	-	84.7	-	-	-	-
Textboxes [78]	93.9	92.0	85.9	91.6	89.7	83.9	
Deep text spotter [13]	92	89	81	89	86	77	9
Li *et al.* [77]	94.2	92.4	88.2	91.1	89.8	84.6	1.1
FOTS* [87]	95.94	93.90	87.76	91.99	90.11	84.77	11.2
He *et al.* [50]	93	92	87	91	89	86	-
Mask TextSpotter [95]	92.5	92.0	88.2	92.2	91.1	86.5	4.8

OCR is combined with a translation model. This can be extremely helpful and time-saving as people travel or consult documents written in foreign languages. Google's Translate application[39] can perform such instant translation. A similar application is instant text-to-speech equipped with OCR, which can help those with visual disability and those who are illiterate [3].

Intelligent Content Analysis OCR also allows the industries 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[40] and Niconico[41]). On the one hand, such extracted text can be used in automatic content tagging and recommendation system. 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 administrator can impose supervision and filtration for inappropriate and illegal content, such as terrorist advocacy.

7 Discussion

7.1 Status Quo

The past several years have witnessed the significant development of algorithms for text detection and recognition. As deep learning rose, the methodology of research has changed from searching for patterns and features, to architecture designs that takes up challenges one by one. We've seen and recognize how deep learning has resulted in great progress in terms of the performance of the benchmark datasets. Following a number of newly-designed datasets, algorithms aimed at different targets have attracted attention, e.g. for blurred images and irregular text. 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. *bank card*, *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 [116], [180] has also allowed the deployment of large-scale systems [10]. Companies including Google Inc. and Amazon Inc. are providing text extraction APIs.

Despite the success so far, algorithms for text detection and recognition are still confronted with several challenges. While human have barely no difficulties localizing and recognizing text, current algorithms are not designed and trained effortlessly. They have not yet reached human-level performance. Besides, most datasets are monolingual. We have no idea how these models would perform on other languages. What exacerbates it is that, the evaluation metrics we use today may be far from perfect. Under PASCAL evaluation, a detection result which only covers slightly more than half of the text instance would be judged as successful as it passes the IoU threshold of 0.5. Under DetEval, one can manually enlarge the detected area to meet the requirement of pixel recall, as DetEval requires a high pixel recall (0.8) but rather low pixel precision (0.4). Both cases would be judged as failure from oracle's viewpoint, as the former can not retrieve the whole text, while the later encloses too much background. A new and more appropriate evaluation protocol is needed. Finally, few works except for TextSnake [92] have considered the problem of generalization ability across

39. https://translate.google.com/intl/en/about/
40. https://www.bilibili.com
41. www.nicovideo.jp/

datasets. Generalization ability is important as we aim to some application scenarios would require the adaptability to changing 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. However, these scenarios are only represented by different datasets individually. We should expect a more diverse dataset.

7.2 Future Trends

History is a mirror for the future. What we lack today tells us about what we can expect tomorrow.

Diversity among Datasets: More Powerful Model Text detection and recognition is different from generic object detection in the sense that, it's faced with unique challenges. We expect that new datasets aimed at new challenges, as we have seen so far [17], [67], [174], would draw attention to these aspects and solve more real world problems.

Diversity inside Datasets: More Robust Model Despite the success we've seen so far, current methods are only evaluated on single datasets after being trained on them separately. Tests of authentic generalization are needed, where a single trained model is evaluated on a more diverse held-out set, e.g. a combination of current datasets. Naturally, a new dataset representing several challenges would also provide extra momentum for this field. Evaluation of cross dataset generalization ability is also preferable, where the model is trained only on one dataset and then tested of another, as done in recent work in curved text [92].

Suitable Evaluation Metrics: a Fairer Play As discussed above, an evaluation metric that fits the task more appropriately would be better. Current evaluation metrics (DetEval and PASCAL-Eval) are inherited from the more generic task of object detection, where detection results are all represented in rectangular bounding boxes. However, in text detection and recognition, the shapes and orientations matter. Tighter and noiseless bounding region would also be more friendly to recognizers. Neglecting some parts in object detection may be acceptable as it remains semantically the same, but it would be disastrous for the final text recognition results as some characters may be missing, resulting in different words.

Towards Stable Performance: as Needed in Security As we have seen work that breaks sequence modeling methods [173] and attacks that interfere image classification models [141], we should pay more attention to potential security risks. Especially, text detection and recognition methods themselves are applied in security services e.g. identity check.

REFERENCES

[1] Icdar 2015 robust reading competition (presentation). http://rrc.cvc.uab.es/files/Robust_Reading_2015_v02.pdf. Accessed: 2018-07-30.

[2] Project gutenberg for ditigizing books. https://www.gutenberg.org. Accessed: 2018-08-08.

[3] Screen reader. https://en.wikipedia.org/wiki/Screen_reader#cite_note-Braille_display-2. Accessed: 2018-08-09.

[4] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. Word spotting and recognition with embedded attributes. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2552–2566, 2014.

[5] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR 2015*, 2014.

[7] Fan Bai, Zhanzhan Cheng, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Edit probability for scene text recognition. In *CVPR 2018*, 2018.

[8] Christian Bartz, Haojin Yang, and Christoph Meinel. See: Towards semi-supervised end-to-end scene text recognition. *arXiv preprint arXiv:1712.05404*, 2017.

[9] Alessandro Bissacco, Mark Cummins, Yuval Netzer, and Hartmut Neven. Photoocr: Reading text in uncontrolled conditions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 785–792, 2013.

[10] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. 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*, pages 71–79. ACM, 2018.

[11] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451, 2017.

[12] Michal Busta, Lukas Neumann, and Jiri Matas. Fastext: Efficient unconstrained scene text detector. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1206–1214, 2015.

[13] Michal Busta, Lukas Neumann, and Jiri Matas. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *Proc. ICCV*, 2017.

[14] Xilin Chen, Jie Yang, Jing Zhang, and Alex Waibel. Automatic detection and recognition of signs from natural scenes. *IEEE Transactions on image processing*, 13(1):87–99, 2004.

[15] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5086–5094. IEEE, 2017.

[16] Zhanzhan Cheng, Xuyang Liu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Arbitrarily-oriented text recognition. *CVPR2018*, 2017.

[17] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 935–942. IEEE, 2017.

[18] MM Aftab Chowdhury and Kaushik Deb. Extracting and segmenting container name from container images. *International Journal of Computer Applications*, 74(19), 2013.

[19] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J Wu, and Andrew Y Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 440–445. IEEE, 2011.

[20] Yuchen Dai, Zheng Huang, Yuting Gao, and Kai Chen. Fused text segmentation networks for multi-oriented scene text detection. *arXiv preprint arXiv:1709.03272*, 2017.

[21] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[22] Deng Dan, Liu Haifeng, Li Xuelong, and Cai Deng. Pixellink: Detecting scene text via instance segmentation. In *Proceedings of AAAI, 2018*, 2018.

[23] Guilherme N DeSouza and Avinash C Kak. Vision for mobile robot navigation: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 24(2):237–267, 2002.

[24] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014.

[25] Yuval Dvorin and Uzi Ezra Havosha. Method and device for instant translation, June 4 2009. US Patent App. 11/998,931.

[26] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2963–2970. IEEE, 2010.

[27] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[28] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61(1):55–79, 2005.

[29] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[30] Yunze Gao, Yingying Chen, Jinqiao Wang, and Hanqing Lu. Reading scene text with attention convolutional sequence modeling. *arXiv preprint arXiv:1709.04303*, 2017.

[31] Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 123–135, 2017.

[32] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252, 2017.

[33] Suman K Ghosh, Ernest Valveny, and Andrew D Bagdanov. Visual attention models for scene text recognition. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on. IEEE, 2017*, volume 1, pages 943–948, 2017.

[34] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[35] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 580–587, 2014.

[36] Lluis Gomez and Dimosthenis Karatzas. Object proposals for text extraction in the wild. In *13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 206–210. IEEE, 2015.

[37] Lluís Gómez and Dimosthenis Karatzas. Textproposals: a text-specific selective search algorithm for word spotting in the wild. *Pattern Recognition*, 70:60–74, 2017.

[38] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[39] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

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

[41] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[42] Alex Graves, Marcus Liwicki, Horst Bunke, Jürgen Schmidhuber, and Santiago Fernández. Unconstrained on-line handwriting recognition with recurrent neural networks. In *Advances in neural information processing systems*, pages 577–584, 2008.

[43] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2315–2324, 2016.

[44] Young Kug Ham, Min Seok Kang, Hong Kyu Chung, Rae-Hong Park, and Gwi Tae Park. Recognition of raised characters for automatic classification of rubber tires. *Optical Engineering*, 34(1):102–110, 1995.

[45] Dafang He, Xiao Yang, Wenyi Huang, Zihan Zhou, Daniel Kifer, and C Lee Giles. Aggregating local context for accurate scene text detection. In *Asian Conference on Computer Vision*, pages 280–296. Springer, 2016.

[46] Dafang He, Xiao Yang, Chen Liang, Zihan Zhou, Alexander G Ororbia, Daniel Kifer, and C Lee Giles. Multi-scale fcn with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 474–483. IEEE, 2017.

[47] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[49] Pan He, Weilin Huang, Tong He, Qile Zhu, Yu Qiao, and Xiaolin Li. Single shot text detector with regional attention. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[50] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5020–5029, 2018.

[51] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Deep direct regression for multi-oriented scene text detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[52] Zhiwei He, Jilin Liu, Hongqing Ma, and Peihong Li. A new automatic extraction method of container identity codes. *IEEE Transactions on intelligent transportation systems*, 6(1):72–78, 2005.

[53] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[54] Michal Hradiš, Jan Kotera, Pavel Zemcík, and Filip Šroubek. Convolutional neural networks for direct text deblurring. In *Proceedings of BMVC*, volume 10, 2015.

[55] Han Hu, Chengquan Zhang, Yuxuan Luo, Yuzhuo Wang, Junyu Han, and Errui Ding. Wordsup: Exploiting word annotations for character based text detection. In *Proceedings of the IEEE International Conference on Computer Vision. 2017.*, 2017.

[56] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[57] Weilin Huang, Zhe Lin, Jianchao Yang, and Jue Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1241–1248, 2013.

[58] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep structured output learning for unconstrained text recognition. *ICLR2015*, 2014.

[59] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

[60] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2016.

[61] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[62] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *In Proceedings of European Conference on Computer Vision (ECCV)*, pages 512–528. Springer, 2014.

[63] Anil K Jain and Bin Yu. Automatic text location in images and video frames. *Pattern recognition*, 31(12):2055–2076, 1998.

[64] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2cnn: rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017.

[65] Le Kang, Yi Li, and David Doermann. Orientation robust text line detection in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4034–4041, 2014.

[66] Dimosthenis Karatzas and Apostolos Antonacopoulos. Text extraction from web images based on a split-and-merge segmentation method using colour perception. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 634–637. IEEE, 2004.

[67] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1156–1160. IEEE, 2015.

[68] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere de las Heras. Icdar 2013 robust reading competition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1484–1493. IEEE, 2013.

[69] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 3, 2017.

[70] Vijeta Khare, Palaiahnakote Shivakumara, Paramesran Raveendran, and Michael Blumenstein. A blind deconvolution model for scene text detection and recognition in video. *Pattern Recognition*, 54:128–148, 2016.

[71] Kye-Hyeon Kim, Sanghoon Hong, Byungseok Roh, Yeongjae Cheon, and Minje Park. PVANET: deep but lightweight neural networks for real-time object detection. *arXiv:1608.08021*, 2016.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[73] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

[75] Jung-Jin Lee, Pyoung-Hean Lee, Seong-Whan Lee, Alan Yuille, and Christof Koch. Adaboost for text detection in natural scene. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 429–434. IEEE, 2011.

[76] Seonghun Lee and Jin Hyung Kim. Integrating multiple character proposals for robust scene text extraction. *Image and Vision Computing*, 31(11):823–840, 2013.

[77] Hui Li, Peng Wang, and Chunhua Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[78] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *AAAI*, pages 4161–4167, 2017.

[79] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5909–5918, 2018.

[80] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[81] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5162–5170, 2015.

[82] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *In Proceedings of European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.

[83] Wei Liu, Chaofeng Chen, and KKY Wong. Char-net: A character-aware neural network for distorted scene text recognition. In *AAAI Conference on Artificial Intelligence*. New Orleans, Louisiana, USA, 2018.

[84] Wei Liu, Chaofeng Chen, Kwan-Yee K Wong, Zhizhong Su, and Junyu Han. Star-net: A spatial attention residue network for scene text recognition. In *BMVC*, volume 2, page 7, 2016.

[85] Xiaoqing Liu and Jagath Samarabandu. An edge-based text region extraction algorithm for indoor mobile robot navigation. In *Mechatronics and Automation, 2005 IEEE International Conference*, volume 2, pages 701–706. IEEE, 2005.

[86] Xiaoqing Liu and Jagath K Samarabandu. A simple and fast text localization algorithm for indoor mobile robot navigation. In *Image Processing: Algorithms and Systems IV*, volume 5672, pages 139–151. International Society for Optics and Photonics, 2005.

[87] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. *CVPR2018*, 2018.

[88] Yuliang Liu and Lianwen Jin. Deep matching prior network: Toward tighter multi-oriented text detection. 2017.

[89] Zichuan Liu, Yixing Li, Fengbo Ren, Hao Yu, and Wangling Goh. Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network. *AAAI*, 2018.

[90] Zichuan Liu, Guosheng Lin, Sheng Yang, Jiashi Feng, Weisi Lin, and Wang Ling Goh. Learning markov clustering networks for scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6936–6944, 2018.

[91] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[92] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *In Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[93] Simon M Lucas. Icdar 2005 text locating competition results. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 80–84. IEEE, 2005.

[94] Simon M Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, and Robert Young. Icdar 2003 robust reading competitions. In *null*, page 682. IEEE, 2003.

[95] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *In Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[96] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. Multi-oriented scene text detection via corner localization and region segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*, 2018.

[97] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. In *IEEE Transactions on Multimedia, 2018*, 2017.

[98] Abdelhamid Mammeri, Azzedine Boukerche, et al. Mser-based text detection and communication algorithm for autonomous vehicles. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 1218–1223. IEEE, 2016.

[99] Abdelhamid Mammeri, El-Hebri Khiari, and Azzedine Boukerche. Road-sign text recognition architecture for intelligent transportation systems. In *2014 IEEE 80th Vehicular Technology Conference (VTC Fall)*, pages 1–5. IEEE, 2014.

[100] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[101] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[102] Anand Mishra, Karteek Alahari, and CV Jawahar. An mrf model for binarization of natural scene text. In *ICDAR-International Conference on Document Analysis and Recognition*. IEEE, 2011.

[103] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC-British Machine Vision Conference*. BMVA, 2012.

[104] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 1454–1459. IEEE, 2017.

[105] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[106] Luka Neumann and Jiri Matas. On combining multiple segmentations in scene text recognition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 523–527. IEEE, 2013.

[107] Lukas Neumann and Jiri Matas. A method for text localization and recognition in real-world images. In *Asian Conference on Computer Vision*, pages 770–783. Springer, 2010.

[108] Lukáš Neumann and Jiří Matas. Real-time scene text localization and recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3538–3545. IEEE, 2012.

[109] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. pages 1520–1528, 2015.

[110] Shigueo Nomura, Keiji Yamanaka, Osamu Katai, Hiroshi Kawakami, and Takayuki Shiose. A novel adaptive morphological approach for degraded character image segmentation. *Pattern Recognition*, 38(11):1961–1975, 2005.

[111] Christopher Parkinson, Jeffrey J Jacobsen, David Bruce Ferguson, and Stephen A Pombo. Instant translation system, November 29 2016. US Patent 9,507,772.

[112] Andrei Polzounov, Artsiom Ablavatski, Sergio Escalera, Shijian Lu, and Jianfei Cai. Wordfence: Text detection in natural images with border awareness. *ICIP/ICPR*, 2017.

[113] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 569–576, 2013.

[114] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[115] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[116] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[117] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048, 2014.

[118] Jose A Rodriguez-Serrano, Albert Gordo, and Florent Perronnin. Label embedding: A frugal baseline for text recognition. *International Journal of Computer Vision*, 113(3):193–207, 2015.

[119] Jose A Rodriguez-Serrano, Florent Perronnin, and France Meylan. Label embedding for text recognition. In *Proceedings of the British Machine Vision Conference*. Citeseer, 2013.

[120] Li Rong, En MengYi, Li JianQiang, and Zhang HaiBin. weakly supervised text attention network for generating text proposals in scene images. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 324–330. IEEE, 2017.

[121] Xuejian Rong, Chucai Yi, and Yingli Tian. Unambiguous text localization and retrieval for cluttered scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3279–3287. IEEE, 2017.

[122] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Springer International Publishing, 2015.

[123] Partha Pratim Roy, Umapada Pal, Josep Llados, and Mathieu Delalandre. Multi-oriented and multi-sized touching character segmentation using dynamic programming. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 11–15. IEEE, 2009.

[124] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[125] Georg Schroth, Sebastian Hilsenbeck, Robert Huitl, Florian Schweiger, and Eckehard Steinbach. Exploiting text-related features for content-based image retrieval. In *2011 IEEE International Symposium on Multimedia*, pages 77–84. IEEE, 2011.

[126] Ruth Schulz, Ben Talbot, Obadiah Lam, Feras Dayoub, Peter Corke, Ben Upcroft, and Gordon Wyeth. 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)*, pages 1100–1105. IEEE, 2015.

[127] Asif Shahab, Faisal Shafait, and Andreas Dengel. Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1491–1496. IEEE, 2011.

[128] Zhang Sheng, Liu Yuliang, Jin Lianwen, and Luo Canjie. Feature enhancement network: A refined scene text detector. In *Proceedings of AAAI, 2018*, 2018.

[129] Karthik Sheshadri and Santosh Kumar Divvala. Exemplar driven character recognition in the wild. In *BMVC*, pages 1–10, 2012.

[130] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[131] Baoguang Shi, Xiang Bai, and Cong Yao. 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, 2017.

[132] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4168–4176, 2016.

[133] Baoguang Shi, Mingkun Yang, XingGang Wang, Pengyuan Lyu, Xiang Bai, and Cong Yao. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 31(11):855–868, 2018.

[134] Baoguang Shi, Cong Yao, Minghui Liao, Mingkun Yang, Pei Xu, Linyan Cui, Serge Belongie, Shijian Lu, and Xiang Bai. Icdar2017 competition on reading chinese text in the wild (rctw-17). In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 1429–1434. IEEE, 2017.

[135] Cunzhao Shi, Chunheng Wang, Baihua Xiao, Yang Zhang, Song Gao, and Zhong Zhang. Scene text recognition using part-based tree-structured character detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2961–2968. IEEE, 2013.

[136] Palaiahnakote Shivakumara, Souvik Bhowmick, Bolan Su, Chew Lim Tan, and Umapada Pal. A new gradient based character segmentation method for video text recognition. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 126–130. IEEE, 2011.

[137] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1):30–34, 1973.

[138] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[139] Bolan Su and Shijian Lu. Accurate scene text recognition based on recurrent neural network. In *Asian Conference on Computer Vision*, pages 35–48. Springer, 2014.

[140] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[141] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[142] Shangxuan Tian, Shijian Lu, and Chongshou Li. Wetext: Scene text detection under weak supervision. In *Proc. ICCV*, 2017.

[143] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *In Proceedings of European Conference on Computer Vision (ECCV)*, pages 56–72. Springer, 2016.

[144] Sam S Tsai, Huizhong Chen, David Chen, Georg Schroth, Radek Grzeszczuk, and Bernd Girod. Mobile visual search on printed documents using text and low bit-rate features. In *18th IEEE International Conference on Image Processing (ICIP)*, pages 2601–2604. IEEE, 2011.

[145] Zhuowen Tu, Yi Ma, Wenyu Liu, Xiang Bai, and Cong Yao. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1083–1090. IEEE, 2012.

[146] Seiichi Uchida. Text localization and recognition in images and video. In *Handbook of Document Image Processing and Recognition*, pages 843–883. Springer, 2014.

[147] Stijn Marinus Van Dongen. *Graph clustering by flow simulation*. PhD thesis, 2000.

[148] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[149] Steffen Wachenfeld, H-U Klein, and Xiaoyi Jiang. Recognition of screen-rendered text. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 1086–1089. IEEE, 2006.

[150] Toru Wakahara and Kohei Kita. Binarization of color character strings in scene images using k-means clustering and support

vector machines. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 274–278. IEEE, 2011.

[151] Cong Wang, Fei Yin, and Cheng-Lin Liu. Scene text detection with novel superpixel based character candidate extraction. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 929–934. IEEE, 2017.

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

[153] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1457–1464. IEEE, 2011.

[154] Kai Wang and Serge Belongie. Word spotting in the wild. In *In Proceedings of European Conference on Computer Vision (ECCV)*, pages 591–604. Springer, 2010.

[155] Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng. End-to-end text recognition with convolutional neural networks. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3304–3308. IEEE, 2012.

[156] Jerod Weinman, Erik Learned-Miller, and Allen Hanson. Fast lexicon-based scene text recognition with sparse belief propagation. In *icdar*, pages 979–983. IEEE, 2007.

[157] Christian Wolf and Jean-Michel Jolion. 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, 2006.

[158] Dao Wu, Rui Wang, Pengwen Dai, Yueying Zhang, and Xiaochun Cao. Deep strip-based network with cascade learning for scene text localization. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 826–831. IEEE, 2017.

[159] Yue Wu and Prem Natarajan. Self-organized text detection with minimal post-processing via border learning. In *Proceedings of the IEEE Conference on CVPR*, pages 5000–5009, 2017.

[160] Chuhui Xue, Shijian Lu, and Fangneng Zhan. Accurate scene text detection through border semantics awareness and bootstrapping. In *In Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[161] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C Lee Giles. Learning to read irregular text with attention mechanisms. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3280–3286, 2017.

[162] Cong Yao, Xiang Bai, and Wenyu Liu. A unified framework for multioriented text detection and recognition. *IEEE Transactions on Image Processing*, 23(11):4737–4749, 2014.

[163] Cong Yao, Xiang Bai, Nong Sang, Xinyu Zhou, Shuchang Zhou, and Zhimin Cao. Scene text detection via holistic, multi-channel prediction. *arXiv preprint arXiv:1606.09002*, 2016.

[164] Cong Yao, Xiang Bai, Baoguang Shi, and Wenyu Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4042–4049, 2014.

[165] Qixiang Ye and David Doermann. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1480–1500, 2015.

[166] Qixiang Ye, Wen Gao, Weiqiang Wang, and Wei Zeng. A robust text detection algorithm in images and video frames. *IEEE ICICS-PCM*, pages 802–806, 2003.

[167] Chucai Yi and YingLi Tian. Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing*, 20(9):2594–2605, 2011.

[168] Fei Yin, Yi-Chao Wu, Xu-Yao Zhang, and Cheng-Lin Liu. Scene text recognition with sliding convolutional character models. *arXiv preprint arXiv:1709.01727*, 2017.

[169] Xu-Cheng Yin, Wei-Yi Pei, Jun Zhang, and Hong-Wei Hao. Multi-orientation scene text detection with adaptive clustering. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1930–1937, 2015.

[170] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):970–983, 2014.

[171] Xu-Cheng Yin, Ze-Yu Zuo, Shu Tian, and Cheng-Lin Liu. Text detection, tracking and recognition in video: A comprehensive survey. *IEEE Transactions on Image Processing*, 25(6):2752–2773, 2016.

[172] Tai-Ling Yuan, Zhe Zhu, Kun Xu, Cheng-Jun Li, and Shi-Min Hu. Chinese text in the wild. *arXiv preprint arXiv:1803.00085*, 2018.

[173] Xiaoyong Yuan, Pan He, and Xiaolin Andy Li. Adaptive adversarial attack on scene text recognition. *arXiv preprint arXiv:1807.03326*, 2018.

[174] Liu Yuliang, Jin Lianwen, Zhang Shuaitao, and Zhang Sheng. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*, 2017.

[175] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional networks. In *In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2528–2535, 2010.

[176] Fangneng Zhan, Shijian Lu, and Chuhui Xue. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. 2018.

[177] DongQuin Zhang and Shih-Fu Chang. A bayesian framework for fusing multiple word knowledge models in videotext recognition. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2003.

[178] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4159–4167, 2016.

[179] Zhou Zhiwei, Li Linlin, and Tan Chew Lim. Edge based binarization for video text images. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 133–136. IEEE, 2010.

[180] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: An efficient and accurate scene text detector. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[181] Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Oriented response networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4961–4970. IEEE, 2017.

[182] Anna Zhu, Renwu Gao, and Seiichi Uchida. Could scene context be beneficial for scene text detection? *Pattern Recognition*, 58:204–215, 2016.

[183] Xiangyu Zhu, Yingying Jiang, Shuli Yang, Xiaobing Wang, Wei Li, Pei Fu, Hua Wang, and Zhenbo Luo. Deep residual text detection network for scene text. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on. IEEE, 2017*, volume 1, pages 807–812, 2017.

[184] Yingying Zhu, Cong Yao, and Xiang Bai. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, 10(1):19–36, 2016.

[185] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *In Proceedings of European Conference on Computer Vision (ECCV)*, pages 391–405. Springer, 2014.