Review of Scene Text Detection and Recognition

Article *in* Archives of Computational Methods in Engineering · January 2019

DOI: 10.1007/s11831-019-09315-1

3 authors, including:

Han Lin
Nanjing Audit University
33 PUBLICATIONS **372** CITATIONS

SEE PROFILE

Archives of Computational Methods in Engineering
https://doi.org/10.1007/s11831-019-09315-1

ORIGINAL PAPER

Review of Scene Text Detection and Recognition

Han Lin[1] · **Peng Yang**[1,2] · **Fanlong Zhang**[1]

Received: 9 October 2018 / Accepted: 8 January 2019

Abstract
Scene texts contain rich semantic information which may be used in many vision-based applications, and consequently detecting and recognizing scene texts have received increasing attention in recent years. In this paper, we first introduce the history and progress of scene text detection and recognition, and classify conventional methods in detail and point out their advantages as well as disadvantages. After that, we study these methods and illustrate the corresponding key issues and techniques, including loss function, multi-orientation, language model and sequence labeling. Finally, we describe commonly used benchmark datasets and evaluation protocols, based on which the performance of representative scene text detection and recognition methods are analyzed and compared.

1 Introduction

Texts in scene image contain high-level important semantic information, which is help to analyzing and understanding the corresponding environment. With the rapid popularization of smart phones and mobile computing devices, images with text data are acquired more conveniently and efficiently. Therefore, scene text recognition (STR) has become active research topic in computer vision, and its related applications are including image retrieval, automatic navigation and human–computer interaction, etc. [1–3]. Moreover, the International Conference on Document Analysis and Recognition (ICDAR) initiates "Robust Reading" competition in 2003, and since then numerous techniques and methods have been proposed to greatly advance the development of STR.

Text detection and recognition are two fundamental tasks for STR. Text detection aims to determine the position of text from input image, and the position is often represented by a bounding box. Generally, the shape of target bounding box may be rectangle, oriented rectangle or quadrilateral. More precisely, parameters (x, y, w, h), (x, y, w, h, θ) and $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ can be used to denotes horizontal, rotated and arbitrary quadrilateral bounding box respectively. Text recognition aims to convert image regions containing text into machine-readable strings. Different from the general image classification, the dimension of output sequence for text recognition is not fixed. In most cases, text detection is a preliminary step of text recognition. Recently, many researchers begin to integrate the detection and recognition tasks into an end-to-end text recognition system. Considering a small lexicon, word spotting offers an effective strategy for realizing end-to-end recognition [4].

The target of traditional optical character recognition (OCR) is mainly document images acquired by scanner [5]. Since even old scanners have enough resolution for text image acquisition, the recognition rates of many OCR methods can easily reach 99%. Compared to traditional OCR, however, STR is more challenging, which are discussed as follows:

(1) Texts are often scattered in the scene image, and there is no prior information about their location. For scanned documents, the number of text lines, line spacing and even the number of words are known. For scene texts, however, we cannot directly apply segmentation methods for document images since there is no such formatting rule.
(2) Scene texts often have variety of sizes, fonts and orientations. Targets in scene image may contain decorated or specially-designed characters, such as presentation slides on screen, calligraphic slogans on wall, and messages on digital signboard. Such texts with multifari-

✉ Peng Yang
llylab@21cn.com

[1] School of Information Engineering, Nanjing Audit University, Jiangshu 211815, China

[2] School of Information Engineering, Nanchang Hangkong University, Jiangxi 330063, China

Published online: 11 January 2019

ous appearance are difficultly recognized by traditional OCR engines.

(3) The quality of scene image acquired by digital devices is potentially poor. At present, scene text covers wide range of applications linked to wearable cameras or massive urban captures which are difficult or undesirable to control. Therefore, characters and their background often have very low contrast or perspective distortion, which results in difficulty for localization and recognition. Figure 1 shows some examples of scene text images that are not easily detected and recognized.

(4) There are many character-like patterns (non-character) in scene image. Since the background of scene image is often complex, there are many ambiguous objects such as leaves, windows or icons that are much like characters or words. Moreover, sometimes scene texts connect to other objects, which easily results in confusing patterns.

In this paper, we mainly provide a comprehensive review about scene text detection and recognition research over the past decade, and highlight the key techniques. Moreover, we compare state-of-the-art methods and report the corresponding performance on several standard benchmark datasets.

2 Scene Text Detection

As mentioned above, scene text detection is a challenging problem. Similar to majority of computer vision tasks, most previous text detection methods are based on handcraft features as well as prior knowledge, and since around 2015 deep learning based methods emerge and gradually become the mainstream.

2.1 Hand-Crafted Feature Extraction Stage

Traditional text detectors focus on developing hand-crafted low-level features to discriminate text and non-text components in scene image, which can be mainly classified into two categories, i.e., sliding window (SW) and connected component (CC) based methods.

2.1.1 SW Methods

SW methods first detect text information by moving a multi-scale sub-window through all possible locations in an image, and then use a pre-trained classifier to identify whether text is contained within the sub-window [22].

Wang et al. [6] provided an end-to-end pipeline for STR, where they perform multi-scale character detection via SW classification. Features are first extracted by chosen entries in a HOG descriptor computed at the window location. Then Random Ferns is applied to evaluate the likelihood of character in the window location. Pan et al. [7] estimated the text existing confidence and scale information via SW. After that, a conditional random field (CRF) model is proposed to filter out the non-text components. Similarly, Mishra et al. [8] used a standard SW method with character aspect ratio prior to detect potential locations of characters in scene image. Wang et al. [9] applied a convolutional neural network (CNN) model with SW scheme to obtain candidate lines of text in given image, and thus estimate text locations. Jaderberg et al. [10] also applied CNN in SW fashion to compute text saliency map, which stays the same resolution as the original image through zero-padding. After that word bounding boxes can be generated based on these saliency maps.

Fig. 1 Examples of scene text images

The main difficulties for this group of methods lie in designing discriminative features to train a powerful classifier, and reasonably managing the number of scanning windows to reduce computation complexity.

2.1.2 CC Methods

CC methods first extract candidate components from the image, and then filter out non-text components using manually designed rules or automatically trained classifiers [23]. Compared to SW methods, such methods are more efficient and robust. There are two representative methods, i.e., stroke width transform (SWT) and maximally stable extremal regions (MSER).

Epshtein et al. [11] presented SWT operator to compute the width of the most likely stroke for image pixel. Canny edge detector is first used to find edges in image. After all the edge pixels in the opposite gradient direction being found, strokes are considered effective and these pixels are grouped into character candidates. Neumann et al. [12] gave a description for character detection problem, i.e., finding all contiguous regions in image such that probability that the sequence represents text has a local maximum. Based on the description, MSER classifier is trained to find region containing characters. Finally, post-processing and connection rules are applied to combine the candidate characters into text line. MSER method needs less priori knowledge and is more robust to language and oriented text. In order to address problems on blurry images or characters with low contrast, the same authors implemented character detection in all extremal regions (ERs) instead of just in MSERs [13, 14]. They use incrementally computable descriptors as features to train a sequential classifier, which can reduce the high false positive rate in real-time. Yin et al. [15] proposed a fast MSERs pruning algorithm, which can significantly reduce the number of character candidates to be processed. Character candidates are clustered into text candidates by the single-link clustering algorithm, whose distance weights and clustering threshold can be automatically learnt. Such new MSER based method is more robust and efficient for text detection.

Generally speaking, CC methods easily bring with numerous non-text components. Therefore, correctly filtering out the false positives is critical to the success of this group of methods.

2.1.3 Hybrid Methods

In order to more efficiently handle scene text with cluttered background information, several hybrid methods are proposed, which make use of the advantages of different methods and combine with specific schemes.

Huang et al. [16] applied CNN to learn high-level features from the MSREs components in image. These components show high discriminant ability and strong robustness against complicated background ones. Moreover, SW model and non-maximal suppression (NMS) are incorporated in the CNN classifier so as to handle the problem of multiple characters connection. Gomez et al. [17] used the MSER algorithm to firstly obtain the initial segmentation of image. After that they propose a text specific selective search strategy, which can group the initial regions by agglomerative clustering in a hierarchy where each node defines a possible word hypothesis. Finally a ranked list of proposals prioritizing the best hypotheses is provided for text detection. Busta et al. [18] proposed a stroke detector, which first finds stroke key-points and then uses them to obtain stroke segmentations for scene text. They show that compared to the traditional MSER method, using stroke specific key-points could detect more characters with less region segmentations. Cho et al. [20] presented Canny text detector using multi-stage algorithm. ER method is first utilized to extract character candidates as many as possible, and the overlapped candidates are eliminated by NMS. After that, the candidates are classified as strong text, weak text or non-text with double threshold. Besides strong text, candidates with low confidence, i.e. weak text, are selected by hysteresis. Finally, the surviving text candidates are grouped to compose sentence. Fabrizio et al. [21] presented a hybrid text detector, which adopts CC method to generate text candidates and also applies texture analysis to compose text string or discard false positives. CCs in image can be first obtained by employing the toggle mapping morphological segmentation (TMMS) algorithm. A shape descriptor based on fast wavelet decomposition is used to classify each CC as character or non- character. After that, a series of texture features are used to train a support vector machine (SVM) for post-processing. He et al. [22] developed contrast-enhancement maximally stable extremal regions (CE-MSERs) detector, which extends the conventional MSERs by enhancing intensity contrast between text patterns and background. Furthermore, they trained a text-attentional CNN that could extract high-level features including text region mask, character label, and binary text/non-text information. The two schemes are incorporated to form an effective text detection model. Zhang et al. [19] proposed a text detector which exploits the symmetry property of character groups. Different from traditional methods that mainly exploit the properties of single characters or strokes, this new detector could utilize context information from scene image to implement text lines extraction.

2.2 Deep learning Era

Recently, deep learning has been widely used in semantic segmentation and general object detection, and achieved

great success. Accordingly, related methods are also being adopted in the field of text detection. In general, semantic segmentation based detectors first extract text blocks from the segmentation map generated by fully convolutional network (FCN). After that, bounding boxes of text are obtained by complex post-processing. General object detectors, however, predict candidate bounding boxes directly by regarding texts as objects. Different from common objects, texts have clear definition of orientation, which should be predicted in addition to the axis-aligned bounding box information.

2.2.1 Semantic Segmentation Based Methods

Yao et al. [24] take scene text detection as a semantic segmentation problem. They use a FCN model based on holistically-nested edge detection (HED) to produce global maps, including information of text region, individual characters and their relationship. And the proposed algorithm could detect multi-oriented and curved texts in scene image. He et al. [33] presented the cascaded convolutional text networks (CCTN), which uses two networks to implement coarse-to-fine segmentation for scene image. Note that the coarse network outputs a per-pixel heat-map indicating the location and probability of text instance, and the fine network outputs two heat-maps for final text detection. Zhang et al. [25] also implement text detection with coarse-to-fine procedure. They first use a FCN (called Text-Block FCN) to predict the salient map of text blocks. After that MSER method is applied to extract multi-oriented text line candidates. Finally, they train another smaller FCN (called Character-Centroid FCN) to provide the character centroid information, based on which false text line candidates can be eliminated. Qin et al. [26] proposed a text detector based on the cascade of two CNNs. Text regions of interest are first produced by a FCN and then resized to a square shape with fixed size. The next stage is the word detection procedure, i.e., training a YOLO-like network to generate oriented rectangular bounding boxes for all words. Finally, a NMS stage is implemented to handle overlapping bounding boxes. He et al. [40] proposed a FCN architecture for multi-oriented scene text detection with two tasks. The classification task implements down-sampled segmentation between text and non-text for input image, and the regression task determines the vertex coordinates of quadrilateral text boundaries through direct regression. Zhou et al. [44] also proposed a FCN based model for scene text detection. Multiple channels of pixel-level text score map and geometry could be generated in this model, which is flexible to produce either word level or line level predictions. Furthermore, a locality aware NMS with low time complexity is proposed for post-processing. Dai et al. [27] presented a detector based on fused text segmentation networks. Features of each image are first extracted through a resnet-101 backbone, and then multi-level feature maps are combined and fed to the region proposed network (RPN) for text region of interest (ROI) generation. The whole architecture could implement text detection and segmentation simultaneously and provide predictions both in the pixel and word level. Deng et al. [28] proposed a scene text detector (called PixelLink) based on instance segmentation. The Single-Shot Detector (SSD) [29] like architecture is used to extract features and perform text/non-text prediction as well as link prediction. The predicted positive pixels are joined together into text instances by predicted positive links. Finally, text bounding boxes are generated directly from the segmentation result without location regression. Li et al. [30] proposed the progressive scale expansion network (PSENet) for segmentation-based text detection. In order to handle the closely adjacent text instances, a progressive scale expansion algorithm is presented. Inspired by the idea of breadth first-search, the expansion starts from the pixels of multiple kernels and iteratively merges the adjacent text pixels until the largest kernels are explored. Yang et al. [31] proposed an IncepText architecture based on instance-aware segmentation, which could deal with scene texts with large variance of scale, aspect ratio, and orientation. ResNet-50 module is first used for feature extraction, and Inception-Text module is appended after feature fusion. Furthermore, deformable PSROI pooling [32] is applied to detect multi-oriented text.

This group of methods is suitable for handling multi-oriented text in real-world scene image. Once text instances in image are very close to each other, however, simply using text/non-text semantic segmentation is hard to separate them. Therefore, post-processing is often inevitable to improve the performance.

2.2.2 General Object Detection Based Methods

Zhong et al. [34] developed a unified framework (called DeepText) for text detection. An inception-RPN is proposed in the framework, which could achieve a high recall with only hundreds of word region proposals via applying multi-scale sliding windows over the feature maps and designing a set of text characteristic prior bounding boxes with each sliding position. Gupta et al. [35] presented an efficient engine that could generate synthetic scene images with text annotations, and all synthetic images are used to train a fully-convolutional regression network (FCRN) for text detection. Since an extreme variant of Hough voting is adopted in FCRN, all individual predictions could be aggregated across the input image. Tian et al. [36] proposed a connectionist text proposal network (CTPN) to localize scene text. In CTPN, VGG16 backbone is first used for feature extraction, and then a vertical anchor mechanism is developed to predict text locations in a fine scale. Finally, a Bi-directional long short term memory (BLSTM) is applied to

connect the fine scale sequential text proposals. Liao et al. [37] presented an end-to-end trainable scene text detector (called TextBoxes), which is inspired by SSD. Since SSD is general object detector, it cannot be directly applied for text detection. To address the problem, text-box layers are included in the architecture of TextBoxes, which could detect the words with extreme aspect ratios by designing long default boxes and irregular 1*5 convolutional filters. Ma et al. [38] proposed a rotation region proposal networks (RRPN), which is built upon the Faster-RCNN [39] architecture. Since the ground truth (GT) of a text region is represented with 5 tuples (x, y, w, h, θ), where θ is the angle parameter, RRPN could generate inclined proposals with text orientation information. Jiang et al. [41] also proposed a Faster-RCNN based architecture, called rotational region CNN (R^2CNN), for arbitrary-oriented text detection. They point out that using an angle parameter could make the network hard to detect vertical texts. Therefore, the coordinates of the first two vertices in clockwise and the height of the bounding box are used to represent an inclined rectangle in R^2CNN. Liu et al. [42] designed a small set of quadrilateral sliding windows to roughly recall text. In training phase, a shared Monte-Carlo method is proposed to compute overlapping area between GT and sliding window. The sliding window beyond the given overlapping threshold is considered as positive and used to finely localize the text. Shi et al. [43] proposed a novel perspective, i.e., texts are composed of segments and links. A segment is a part of a word or text line, and a link connects two adjacent segments. Both segments and links are detected by a SSD like network, and then they are taken as nodes and edges of a graph respectively. Finally, a depth-first search (DFS) algorithm is performed on the graph to find the connected components (word or text line). Liao et al. [45] presented a rotation-sensitive regression detector (RRD) based on SSD, which has two network branches. The regression branch extracts rotation-sensitive features by rotating the convolutional filters, while the classification branch extracts rotation-invariant features by pooling the rotation-sensitive features.

This kind of detectors is often trained by bounding-box annotations just like general object detection methods do, which is difficult to learn fine information of text. While handling small-scale texts, only using single shot model may result in accuracy loss. Moreover, it requires designing anchors or default boxes with various scales, aspect ratios and orientations in advance.

2.2.3 Hybrid Methods

Recently, some researchers try to combine the two kinds of above methods so as to correctly detect texts under more complex situations. He et al. [46] proposed a text attention model, which encodes strong text-specific information using a pixel-wise text mask. Such model could effectively suppress background interference in the convolutional features. Furthermore, multi-scale inception features are aggregated to encode rich local and context information for text prediction. The whole detector works in a coarse-to-fine manner. Zhong et al. [47] presented an anchor-free region proposal network (AF-RPN), which could generate high-quality inclined text proposals directly without designing complicated hand-crafted anchors. In AF-RPN, three detection modules are attached on different pyramid levels for detecting small, medium and large text instances. Lyu et al. [48] proposed a hybrid network for multi-oriented scene text detection. The corner points of text region are first detected, and at the same time position sensitive segmentation maps are predicted. After that, candidate bounding boxes are generated by sampling and grouping corner points, and finally suppressed by using NMS. He et al. [49] presented an end-to-end text spotter, which is based on the idea of mask R-CNN [50]. Especially, a text-alignment layer is designed by introducing a grid sampling scheme. It aims to compute fixed length convolutional features that precisely align to a detected text region with arbitrary orientation. The bounding box and segmentation mask of text could be jointly predicted in the multi-task model.

3 Discussion

In general, traditional hand-crafted feature extraction based methods consist of several steps, which make the detection system complicated and inefficient, and easily result in error accumulation. Moreover, they need too many manual optimizations of classification rules. Deep learning based methods, however, inherit the merits of machine learning. As long as having sufficient number of training samples, they could outdistance the traditional methods in terms of both accuracy and efficiency. Figure 2 shows the focused scene text detection results on standard datasets (including ICDAR 2003, ICDAR 2005, ICDAR 2011 and ICDAR 2013) in terms of F-measure reported in literatures mentioned in Sects. 2.1 and 2.2. The blue and red bars represent traditional and deep learning based methods respectively. Obviously, deep learning based methods achieve overwhelming performance, which explains why they become the mainstream recently.

4 Scene Text Recognition

Similar to text detection, scene text recognition also experiences the transition from traditional means using handcrafted features to deep learning era. In this section, we roughly classify current mainstream text recognition methods into

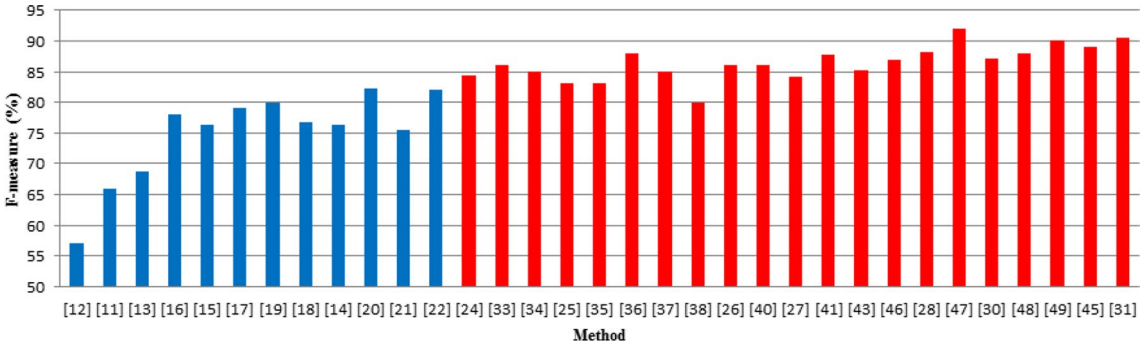


Fig. 2 Performance comparison of representative scene text detectors

three categories: character classification based, word classification based and sequence based methods.

4.1 Character Classification Based Methods

Bissacco et al. [51] use a deep neural network that is trained on HOG features for character classification. In order to enhance the recognition performance, a two-level language model is adopted: a compact character-level n-gram model is held in RAM and a much larger distributed word-level n-gram model is accessed over the network. Jaderberg et al. [57] proposed a CNN based architecture employing a conditional random field (CRF) graphical model. In this model, unary terms are provided by a CNN that predicts characters at each position of the output, and higher order terms are provided by another CNN that detects the presence of n-grams. Lee et al. [60] presented recursive recurrent neural networks (RNNs) with attention model for text recognition. The RNNs could be applied for learning character-level language model without using n-grams. The soft-attention mechanism allows the model to select features flexibly for end-to-end training.

This group of methods finds individual characters in scene image and consequently recognizes them one by one. Complex heuristic rules or language models are often indispensable to integrate characters into words due to the occurrences of missing or superfluous characters.

4.2 Word Classification Based Methods

Jaderberg et al. [52] proposed a synthetic data engine, which could generate plenty of cropped word images with different styles. A CNN framework is trained using synthetic data without handcrafted labeling and achieves high performance for word recognition. Shi et al. [56] presented a variant of CNN for script identification under multilingual scenarios. In this network, feature maps that have a fixed number of rows but a variable number of columns are input to a spatially-sensitive pooling (SSP) layer, which could handle images with arbitrary sizes. Furthermore, a multi-stage pooling scheme is adopted so as to utilize both higher and lower level features for recognition. Kang et al. [63] designed a context-aware convolutional recurrent network for word recognition. Besides a lexicon dictionary, the metadata of the input image, such as title, tags, and comments, are used as a context prior to enhance the recognition rate. Yang et al. [65] proposed an adaptive ensemble of deep neural networks (AdaDNNs), which could select and combine network components at different iterations within a Bayesian-based formulation framework for text recognition.

Word recognition is actually a multi-class classification task with a large number of class labels (e.g. the number of English words is about 90,000). The strong expression and computation ability of CNN make this task possible. However, the deformation of long word image may affect the recognition rate. Furthermore, this kind of methods often relies on a pre-defined dictionary.

4.3 Sequence Based Methods

Shi et al. [55] proposed a convolutional recurrent neural network (CRNN) for image-based sequence recognition. A standard CNN model is first used to extract a sequential feature representation from input image. Then a bidirectional long-short term memory (LSTM) network is connected with the top convolutional layers to predict a label distribution for each frame of feature sequence. Finally, the connectionist temporal classification (CTC) is applied to find the label sequence with the highest probability conditioned on the per-frame predictions. He et al. [58] also developed a deep-text recurrent network (DTRN) for scene text recognition. Similar to [55], a MaxOut CNN is responsible for encoding input image into an ordered sequence, and a LSTM is employed to decode the CNN sequence into a word string. In order to deal with perspective distortion text and curved text, Shi et al. [59] proposed a recognizer with automatic rectification. The input image is first employed thin-plate-spline (TPS) transformation, and then the rectified image is

fed to a sequence recognition network (SRN) to obtain the final result. The methods mentioned above are mainly under an encoder-decoder framework, and use a frame-wise loss to optimize the model. However, the misalignment between the ground truth (GT) sequence and the output probability distribution (PD) sequence may mislead the training [68]. In [68], an edit probability (EP) method is proposed for accurate text recognition. EP measures the probability of a text string conditioned on the input image under parameters for training attention model, meanwhile considering the possible occurrences of missing/superfluous characters.

The advantages and disadvantages of the three kinds of methods for text recognition are summarized in Table 1.

4.4 Hybrid Methods

In this subsection, we also review some hybrid text recognition methods, which mainly rely on intricate graphical model or hand-crafted feature designing, and do not strictly fall into the above categories. Shi et al. [71] use the tree-structured model to generate detection windows that contain candidate characters. Then a CRF model is built on the detection windows to decide character locations. Finally, word recognition is implemented according to a cost function defined by character detection scores, spatial constraints and linguistic knowledge. Yao et al. [72] represent each candidate character by a set of strokelets that could capture the essential substructures of character at multi-scales. Coupled with HOG descriptor, they could train a random forest classifier with high performance and efficiency. Almazan et al. [54] proposed a word recognition method based on embedded attributes. On one hand, a pyramidal histogram of characters (PHOC) representation for each word is defined, which embeds label strings into a d-dimensional space. On the other hand, word image is represented using Fisher vector. Finally, the attributes with PHOCs could be learned by training a SVM. Lou et al. [62] represent word recognition model as a high-order factor graph, where hypothetical neighboring candidate characters are constructed edges of the graph and taken as random variables. Four factors, i.e., transition, smoothness, consistency, and singleton, are defined and applied for word parsing.

4.5 End-to-end Text Spotting

Text detection and recognition are usually combined to implement text spotting, rather than being treated as separate tasks. In a unified system, the recognizer not only produces recognition outputs but also regularizes text detection with its semantic-level awareness [70]. Wang et al. [9] applied CNN to implement end-to-end text recognition. In this model, NMS is used to remove overlapping candidates and obtain the set of line-level bounding boxes for texts. And then beam search technique is used to find the best segmentation of words. The proposed method achieves state-of-the art results under tasks of character recognition, lexicon driven cropped word recognition and end-to-end recognition. Yao et al. [53] presented a unified framework, where text detection and recognition share both features and classification. Furthermore, the dictionary is generated according to Bing search, whose error correction scheme can be used to enhance the recognition rate. Jaderberg et al. [61] also proposed t an end-to-end text spotting system. Word level bounding box proposals are first obtained with high recall, and then filtered by a random forest classifier for improving precision. Two CNNs are used for bounding box regression and text recognition respectively. Moysset et al. [64] designed a CRNN system, in which the convolutional layers share parameters over the different regressors to find text lines locally, and a 2D-LSTM model is trained with CTC alignment to recognize texts. Gomez et al. [67] presented a text-specific proposal method, which first extracts connected components from input image, and then groups them by their similarity via single linkage clustering (SLC). Furthermore, a ranking strategy is designed to prioritize the best word proposals. Finally, an end-to-end word spotting system can be built by incorporating the word recognizers provided in [61]. Liao et al. [70] proposed a novel text detector called TextBoxes ++. TextBoxes ++ is an extension of [37], which could efficiently detect arbitrary-oriented scene text. Combined with a text recognizer, TextBoxes ++ can also be used for end-to-end text spotting.

More recently, researchers begin to design unified end-to-end trainable deep learning network (DNN) that could predict both text regions and text labels in a single forward pass. Bartz et al. [66] presented a single DNN that could

Table 1 Comparison of different kinds of text recognition methods

Method	Strength	Weakness
Character classification based	Be insensitive to font variation, noise, blur and orientation	Rely on complex heuristic rules or language models
Word classification based	Can effectively recognize words in scene image with a large number of class labels	Rely on a lexicon and hardly to handle long word with deformation
Sequence based	Do not rely on the precision of text segmentation, and can process arbitrary strings	Need to design proper objective function to optimize the network parameters

train text detector and recognizer from input image. Moreover, a recurrent spatial transformer is applied as attention mechanism, which makes the localization of the text be learned by the network itself. Liu et al. [69] adopted FCN to find bounding boxes of text, based on which a RoIRotate operator is introduced to extract proper features from shared feature maps. Finally, the features of text proposal are fed to RNN and CTC for text recognition.

5 Key Techniques for Scene Text Detection and Recognition

In this section, state-of-the-art techniques used in current scene text detection and recognition methods are reviewed. As mentioned in Sect. 2, deep learning based methods have become the mainstream for text detection. Therefore, Sects. 4.1 to 4.3 analyze the relevant schemes and issues, including network architecture, loss function and multi-orientation detection. With text recognition, techniques related to language model and sequence labeling are discussed in Sects. 4.4 and 4.5.

5.1 Network Architecture

5.1.1 Fully Convolutional Network (FCN)

FCN [73] could yield hierarchies of features for effective semantic segmentation (see Fig. 3). Since the merits of multi-scale learning and prediction conform to the nature of scene text, many methods [24–26, 33, 40] adopt FCN as their backbone for text detection. Generally, a pixel-wise text/non-text salient map is first obtained by using FCN, which produces pixel-wise labeling or labeled region containing texts. After that, candidate bounding boxes of text could be generated. By applying skip architecture of FCN, receptive fields with different sizes could be helpful to encode both local features and global context of text.

5.1.2 Resnet

Deeper neural networks are more difficult to train, since the accuracy may get saturated and degrade rapidly. To address the degradation problem, He et al. [74] proposed a deep residual learning framework (called Resnet), whose building block is defined as $y = F(X, \{W_i\}) + x$ (see Fig. 4), where x and y are the input and output vectors of the layers considered, and $F(X, \{W_i\})$ is the residual mapping to be learned. Some text detectors [27, 31] use Resnet 50/101 as backbone for feature extraction.

5.1.3 Regions with CNN (R-CNN)

Fast R-CNN [39] is an end-to-end architecture for object detection. In this architecture, an input image and multiple regions of interest (RoIs) are input into a FCN, and softmax probabilities and per-class bounding-box regression offsets are the outputs (see Fig. 5a). Faster R-CNN [76] makes improvement on Fast R-CNN, which aims to reduce the time spending on region proposals generation (see Fig. 5b). A region proposal network (RPN) that shares full-image convolutional features with the detection network is proposed, and the RPN and Fast R-CNN are finally merged into a single network by sharing their convolutional features. By incorporating additional components into these

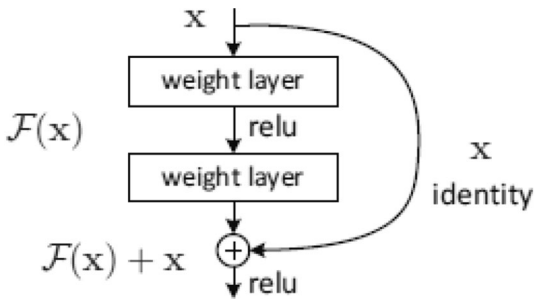
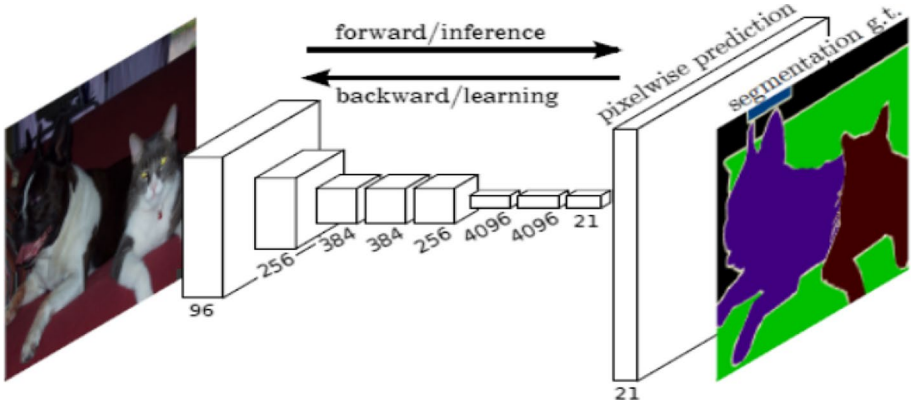
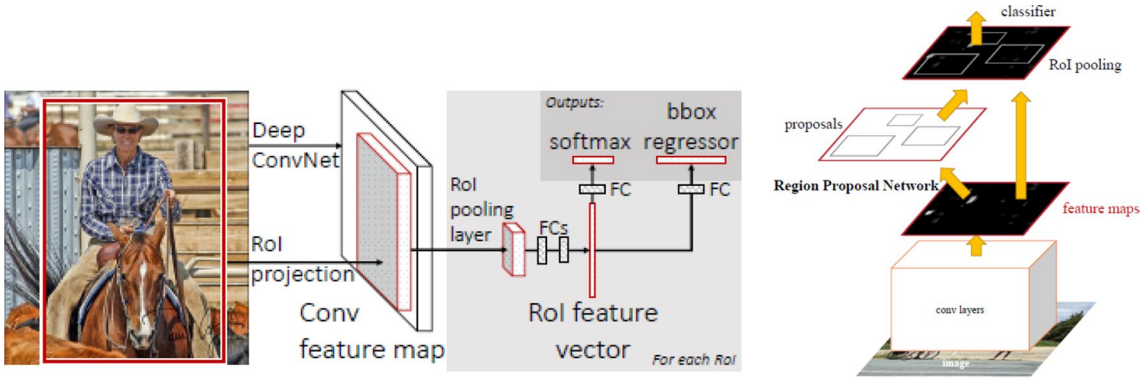


Fig. 4 A building block for residual learning [74]

Fig. 3 Architecture of FCN [73]





(a) Fast R-CNN [39] **(b)** Faster R-CNN [77]

Fig. 5 Architecture of R-CNN series. **a** Fast R-CNN [39], **b** faster R-CNN [76]

architectures, several text detection methods [34, 38, 41, 49] with computational efficiency are proposed.

5.1.4 You only Look Once (YOLO)

YOLO [75] is a single convolutional network that simultaneously predicts multiple bounding boxes and class probabilities for those boxes (see Fig. 6). Since YOLO takes object detection as a single regression problem, it extremely fast comparing with R-CNN based system. However, it may achieve poor precision while localizing objects with small size. Therefore, it cannot be directly applied for text detection. Inspired by YOLO, Gupta et al. [35] proposed a fully-convolutional regression network (FCRN), which could effectively and efficiently detect texts in scene image.

5.1.5 Single Shot Detector (SSD)

SSD [29] defines a set of default boxes for the output space of bounding boxes, and it simultaneously predicts the shape offsets and the confidences for all object categories (see Fig. 7). In SSD, predictions are combined from multiple feature maps with different resolutions. Compared to YOLO, SSD could effectively deal with objects of various sizes. Moreover, SSD eliminates proposal generation and feature resampling, which is different from R-CNN based network. Since SSD integrates the advantages of YOLO and Fast R-CNN/Faster R-CNN, many methods [37, 42, 43, 45, 48] extend this architecture for text detection by giving some specific modifications, such as designing default boxes with larger aspect ratios or multi orientations, and adopting inception-style convolutional filters.

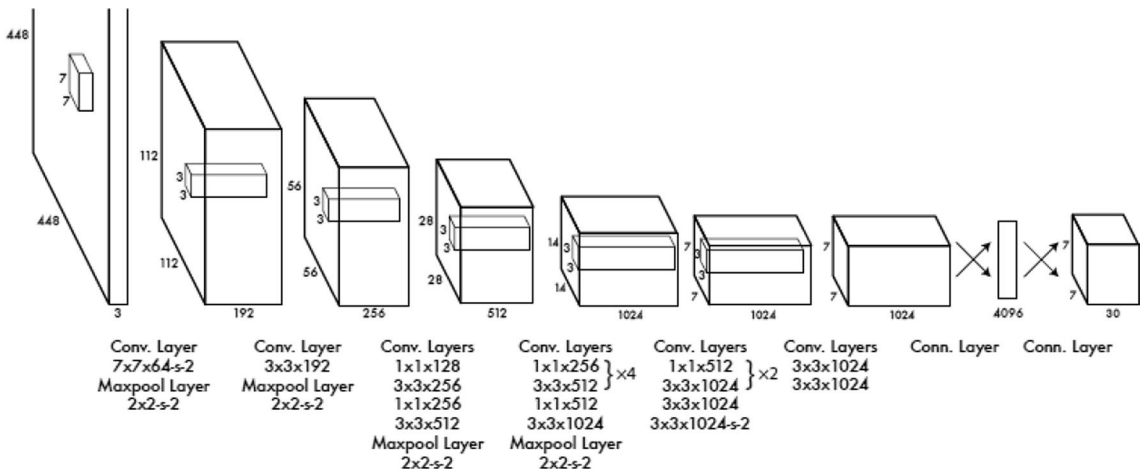


Fig. 6 Architecture of YOLO [75]

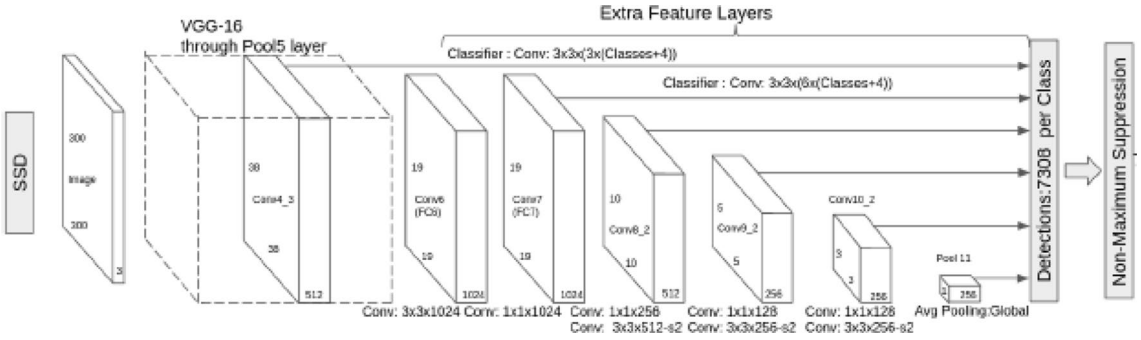


Fig. 7 Architecture of SSD [29]

5.2 Loss Function

Just like in general machine learning model, a loss function should be defined first in deep neural network to measure the gap between prediction and actual value. And then training algorithm seeks to minimize the loss function. The smaller the loss function is, the more robust the model is. Most work often takes text detection as a multi task learning problem, e.g. classification and regression. In this section, some commonly used loss functions for text detection are listed and discussed.

5.2.1 Cross-Entropy Loss Function

It is often used in tasks such as pixel/instance classification or segmentation [25, 27, 28, 30, 31, 33, 44, 48], which is defined as follow

$$L_{ce} = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)] \tag{1}$$

where y_n and $\hat{y}_n$ are actual value and prediction respectively. Note that if the same weight is put on all positive pixels, it may achieve poor performance while handling instances with small areas. Therefore, several balanced cross-entropy losses [28, 44] are also introduced to facilitate the training procedure.

5.3 Softmax Loss Function

It should be found in many general object detection methods, which is defined as follow

$$L_{sm} = \log\left(\sum_{j=0}^{m-1} e^{z_j}\right) - z_y \tag{2}$$

where z_y is the ith value on score vector for classification, and y is the classification label. This function is used in [34, 36–38, 41, 43, 45–47, 49] as the loss for distinguishing text ($y=1$) and non-text ($y=0$).

5.4 Smooth-L1 Loss Function

It is often used for bounding box regression task [27, 31, 34, 36–38, 41, 43–48], which is defined as follow

$$L_{reg} = \sum_{i\in S} smooth_{L1}(p_i, p*) \tag{3}$$

in which,

$$smooth_{L1}(x) = \begin{cases} 0.5(\sigma x)^2 \; if|x| < 1/\sigma^2 \\ |x| - 0.5/\sigma^2 \; otherwise \end{cases} \tag{4}$$

where p and $p*$ are predicted value and ground truth respectively, and x represents the error between p and $p*$. Note that the deviation function of Smooth-L1 is also a piecewise function. In [42], Liu et al. defined a continuous function as follow

$$smooth_{Ln}(x) = (|x|+1)\ln(|x|+1) - |x| \tag{5}$$

They claims that smooth-Ln loss could achieve the tradeoff between robustness and stability (see Fig. 8)

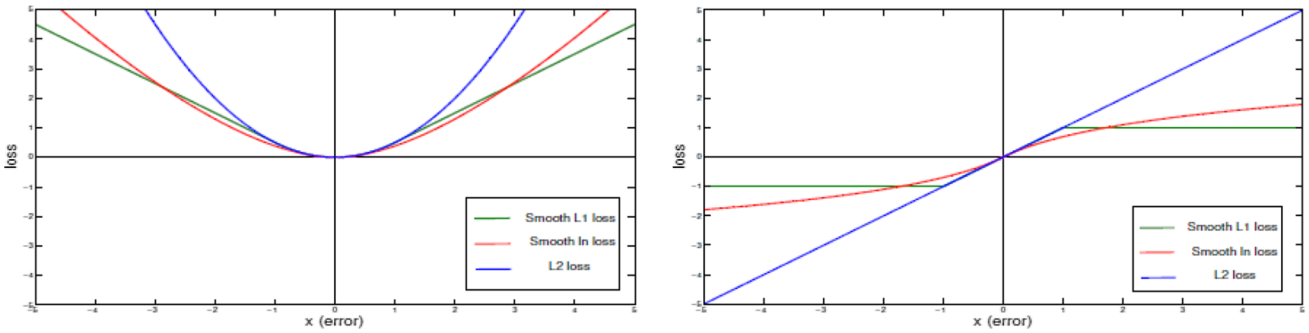
5.4.1 Squared Loss Function

It is a conventional loss for regression task, which is defined as follow

$$L_{squ} = (y-\hat{y})^2 \tag{6}$$

where y and $\hat{y}$ are actual value and prediction respectively. In [26] [35], a bounding box is parameterized in terms of the position of its center, width, height, orientation angle and the confidence that the box contains a word. While training the network, all the parameters are optimized by minimizing a multi-part squared loss function.

There are many other loss functions used for scene text detection. For example, the Dice loss [48] is adopted to



(a) forward loss functions. **(b)** backward deviation functions.

Fig. 8 Comparison of smooth-L1 and smooth-Ln [42]

implement position-sensitive segmentation, and the IOU loss [44] is applied for regressing four channels of axis-aligned bounding box since it is invariant against texts with different scales.

5.5 Multi-orientation Detection

Most of the previous work focuses on horizontal text detection and achieves pretty good performance. However, text in real-world situation could appear with any orientation. Therefore, text orientation needs to be estimated and corrected for subsequent recognition procedure. Although many studies [81–89] have concentrated on multi-oriented scene text detection, the accuracy rates need to be further improved. With the initiating of ICDAR 2015 Competition Challenge 4, a large number of deep learning based methods have stood out, and achieved superior performance over conventional approaches.

In [24], individual characters and their relationship, i.e., linking orientation are considered, and the corresponding prediction maps are produced by training the holistically-nested edge detection (HED) [77] based network. Since HED could find edges of different scales and orientations, it can be used for multi-orientation text detection. Similar work could be found in [43], where the oriented text is decomposed into segments and links, and the final detection results are produced via combining segments connected by links. Since text lines from the same text block often have a roughly uniform spatial layout, a projection profile based skew estimation algorithm [78] is used to determine the possible orientation of text line in [25]. In [27, 33], pixel-wise text region masks with arbitrary shapes are taken as supervision information for training segmentation network so as to handle multi-orientation texts. In [30], the concept "kernel" is introduced, which denotes multiple predicted segmentation areas of text instance. The kernels have the similar shape and locate at the same central point with differ scales. A progressive scale expansion algorithm that could make the kernels grow from small to large scale, is used to obtain the final detections. Therefore, the prediction is robust to arbitrary shapes and orientations. In [31], the position-sensitive RoI (PSROI) pooling [79] is replaced by a deformable PSROI pooling, which could implement multi-oriented text detection through adding offsets to the spatial binning positions.

Note that most of above work includes segmentation step, which is usually time-consuming. A new trend inspired by general object detection has emerged recently, i.e., generating inclined proposals/boxes to roughly recall text, and then implementing bounding box regression to finely localize text region. Text orientation information could be represented by different ways, such as rotation anchors [38], inclined minimum area rectangle [41] or quadrangles inside horizontal sliding windows [42]. Different from previous text detection methods that rely on shared features for both classification and oriented bounding box regression, active rotating filters (ARF) [80] are used to extract rotation-sensitive features in [45]. Since ARF convolves feature map with a canonical filter and its rotated clones, it can help to capture rotation sensitive features. In [48], scene text detection is implemented by localizing corner points of text bounding boxes and segmenting text regions in relative positions (see Fig. 9). The candidate boxes are generated by grouping corner points according to the scores of segmentation maps.

5.6 Language Model

Strong language prior, e.g. probability distribution over character/word sequence, would make major contribution to final text recognition. Some characters or strings cannot be easily distinguished, such as the number "0" and the character "O", or the string "cl" and character "d". If a proper language model is adopted to consider the context information, these cases must be eliminated.

Fig. 9 Corner points and position-sensitive maps prediction [48]

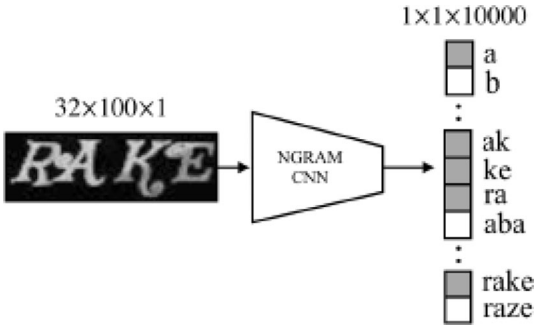


Fig. 10 The N-gram encoding model [57]

Inspired by the successful applying of hidden markov model (HMM) in voice recognition, a hybrid HMM/Maxout architecture is proposed in [90], which could sequence words into their corresponding character/inter-character regions by integrating a lexicon. The method is highly accurate as well as fast, since it takes constant time relative to lexicon size. Conditional random field (CRF) model is adopted to predict character position in [8, 91, 92]. The CRF is defined over a set of random variables, and each random variable denotes a potential character in word. In order to recognize weak character or non-dictionary words, however, it needs to compute unary and higher-order terms for all candidate characters, which results in expensive computation. In [51], the beam search based on n-gram model is used to obtain candidate characters. Beside this language model, a simple dictionary is also maintained for providing a soft scoring signal. Finally, the candidate characters are re-ranked by using both language model and shape model. Similarly, a word is taken as a composition of bag-of-n-grams in [57]. In order to compress encoding representation, the model only selects a subset of the space of all possible n-grams. Since the n-gram based CNN has a large number of output nodes, e.g. 10 k output units for n = 4 (see Fig. 10), it increases the training complexity. Different from the above methods, the recurrent neural network (RNN) is used in [60] to model the character-level statistics for text. In this model, character recognition is considered as a task of learning mappings from pixel intensities to character-level vectors, and does not need n-grams any more.

5.7 Sequence Labeling

As mentioned in Sect. 3.1, many character classification based text recognition methods firstly detect individual characters in image, and sequently recognize each character using CNN models. In order to train a strong character detector, however, we need a large number of labeled character images, which is unrealistic in most cases. Word classification based methods assign a class label to each word, and treat text recognition as an image classification problem. Such methods often train CNN models with a huge number of classes. For English there are about 90 K words, and for Chinese however, the number of potential words may exceed 1 million. Moreover, CNN models are often hard to deal with long words (the number of characters is large). Recently, the state-of-the-art methods consider text spotting as a sequence labeling problem. These methods could generate an ordered high level sequence from input image, and have properties of handling text with arbitrary lengths, lexicon free and avoiding the character segmentation. Some key techniques are reviewed as follows.

5.7.1 Recurrent Neural Network (RNN)

RNN is an important branch of DNN family, which does not need the position information of each element in a sequence image. In [55, 58, 59], a CNN model is first used to convert text image into a sequence of features, and then sequential features are fed to a RNN model for learning context information and generating a predicted sequence. Traditional RNN is hard to transmit the gradient information consistently over long time due to the vanishing gradient problem. The RNN model adopted in [55, 58, 59] is the long-short term memory (LSTM) structure. To be more precisely, two LSTMs, one forward and one backward, are combined into a bidirectional LSTM (see Fig. 11).

5.7.2 Connectionist Temporal Classification (CTC)

In CNN + LSTM model [55, 58], the length of the LSTM outputs may not consistent with that of the target string.

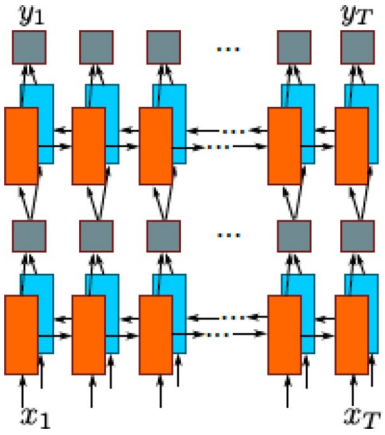


Fig. 11 The structure of deep bidirectional LSTM [55]

Therefore, the CTC [93] is applied to approximately map the LSTM sequential output into its target string:

$$S_w^* \approx \mathrm{B}(\arg\max_{\pi} P(\pi|p)) \tag{7}$$

where B is the projection that removes the repeated labels and the non-character labels.

6 Evaluation and Comparison

Scene text detection and recognition have received increasing attention in computer vision and document analysis, and many approaches and methods have been proposed so far. Therefore, it is impossible to give fair evaluation and comparison for all of them. In this section, we first summarize the widely used datasets and protocols for text detection and recognition. After that, we mainly survey published results of the representative methods for comparison.

6.1 Benchmark Datasets

In this section, we describe the widely used benchmark datasets for tasks of text detection and recognition, whose features are summarized in Table 2.

ICDAR 2003 [94]. It is the first released benchmark for scene text detection and recognition from ICDAR Robust Reading Competition. There are 258 natural images for training and 251 natural images for testing. All the text instances in this dataset are in English and are horizontally placed.

ICDAR 2011 [95]. It inherits from ICDAR 2003 and has made some modification. There are 229 natural images for training and 255 natural images for testing.

ICDAR 2013 [96]. It also inherits from ICDAR 2003 and has made some modification. There are 229 natural images for training and 233 natural images for testing.

ICDAR 2015 [97]. It is from the Incidental Scene Text Challenge of the ICDAR 2015 Robust Reading Competition. The dataset includes 1500 natural images in total, which are acquired using Google Glass. The text instances (annotated by 4 vertices of the quadrangle) are usually skewed or blurred in ICDAR 2015, since they are acquired without user's prior preference or intention.

ICDAR 2017 MLT [98]. It is a large scale multi-lingual text dataset, which is composed of complete scene images with 9 languages. There are 7200 training images, 1800 validation images and 9000 testing images in this dataset.

MSRA-TD500 [99]. It has 500 high resolution natural scene images, where the text instances present with multi orientations and the language types include both Chinese and English. There are 300 images for training and 200 images for testing.

COCO-Text [100]. It is the largest benchmark that could be used for text detection and recognition so far. The original images are from the Microsoft COCO dataset, and 173,589 text instances from 63,686 images are annotated

Table 2 Benchmark datasets for text detection and recognition

Dataset	Annotation	Orientation	Language	Task	End-to-end
ICDAR 2003	Character/word	Horizontal	English	Detection/recognition	Yes
ICDAR 2011	Word	Horizontal	English	Detection/recognition	Yes
ICDAR 2013	Character/word	Horizontal	English	Detection/recognition	Yes
ICDAR 2015 Incidental	Word	Multi oriented	English	Detection/recognition	Yes
ICDAR 2017 MLT	Word	Multi oriented	Multi lingual	Detection/recognition	Yes
MSRA-TD500	Text line	Multi oriented	English/Chinese	Detection	No
COCO-Text	Word	Horizontal	English	Detection/recognition	Yes
SVT	Word	Horizontal	English	Detection/recognition	Yes
RCTW-17	Text line	Multi oriented	Chinese	Detection	Yes
IIIT 5 k	Character/word	Horizontal	English	Recognition	No
SynthText	Character/word	Horizontal	English	Detection/recognition	No
Synth90 k	Word	Horizontal	English	Recognition	No

in COCO-Text. There are 43,686 images for training and 20,000 images for validation/testing.

Street View Text (SVT) [101]. It consists of 350 images annotated with word-level axis-aligned bounding boxes from Google Street View. It contains smaller and lower resolution text, and not all text instances within it are annotated.

RCTW-17 [102]. It contains various kinds of image, including street views, posters, menus, indoor scenes and screenshots for competition on reading Chinese text in image. The dataset contains about 8000 training images and 4000 test images, whose annotations are similar to ICDAR2015.

IIIT 5 k [103]. It contains 5000 cropped word images downloaded from Google image search. There are 2000 images for training and 3000 images for testing. Each image has an associated 50 word lexicon (IIIT5 k-50) and 1 k word lexicon (IIIT5 k-1 k).

SynthText [104]. It contains 858,750 synthetic images, where texts with random colors, fonts, scales and orientations are rendered on natural images carefully to have a realistic look. The texts in this dataset are annotated in character, word and line level.

Synth90 k [105]. It contains about 9 million synthetic cropped word images, and covers 90 k different English words. Similar to SynthText, the synthetic data in Synth90 k is highly realistic. There are approximate 8 million images for training and 900 k images for testing.

6.2 Evaluation Protocols

In this section, we summarize evaluation protocols for text detection and recognition. The task of text detection could be commonly evaluated using ICDAR or DetEval protocol, and the task of text recognition could be commonly evaluated using word recognition accuracy or end-to-end recognition protocol.

6.2.1 ICDAR Detection Protocol

First, the best match $m(r, R)$ for a rectangle r in a set of rectangles R is defined as follow

$$m(r, R) = \max m_p(r, r') | r' \in R \tag{8}$$

where m_p denotes the match between two rectangles of text instances, which can be calculated as the area of intersection divided by the area of the minimum bounding box containing both rectangles. Then, the metrics of precision (P), recall (R) and F-measure(F) can be defined as follows

$$P = \frac{\sum_{r_e \in E} m(r_e, T)}{|E|} \tag{9}$$

$$R = \frac{\sum_{r_t \in T} m(r_t, E)}{|T|} \tag{10}$$

$$F = \frac{1}{\alpha/P + (1-\alpha)/R} \tag{11}$$

where T and E are respectively the sets of ground-truth and estimated rectangles, and r_t and r_e are respectively a ground-truth and an estimated rectangle. α is weight parameter, which is often set to 0.5.

6.2.2 DetEval Detection Protocol

Since standard ICDAR detection protocol is unable to handle the cases of one-to-many and many-to-many matches among the ground truth and detections, it always underestimates the performance of text detection algorithms. To address the problem, Wolf et al. proposed the DetEval protocol to comprise the area overlap and the object level evaluation. In this protocol, the metrics of precision (P') and recall (R') can be defined as follows

$$P' = \frac{\sum_i Match_D(D_i, G, t_r, t_p)}{|D|} \tag{12}$$

$$R' = \frac{\sum_j Match_G(G_j, D, t_r, t_p)}{|D|} \tag{13}$$

where $Match_D$ and $Match_G$ are functions that consider the different types of matches:

$$Match_D(D_i, G, t_r, t_p) = \begin{cases} 1 & \text{if } D_i \text{ matches against a single detected rectangle} \\ 0 & \text{if } D_i \text{ does not match against any detected rectangle} \\ f_{sc}(k) & \text{if } D_i \text{ matches against several } (\rightarrow k) \text{ detected rectangles} \end{cases} \tag{14}$$

$$Match_G(G_j, D, t_r, t_p) = \begin{cases} 1 & \text{if } G_j \text{ matches against a single detected rectangle} \\ 0 & \text{if } G_j \text{ does not match against any detected rectangle} \\ f_{sc}(k) & \text{if } G_j \text{ matches against several } (\rightarrow k) \text{ detected rectangles} \end{cases} \tag{15}$$

where $f_{sc}(k)$ is a parameter function that controls the amount of punishment, and it is often set to 0.8.

6.2.3 Yao's Detection Protocol

While handling texts with arbitrary orientation, the overlap ratio computed in the way of standard ICDAR protocol is possibly not accurate. Therefore, Yao et al. [81] proposed an evaluation protocol that considers true or false positives based on the overlap ratio between the estimated minimum area rectangles and the ground truth rectangles. If the included angle between the estimated rectangle and the ground truth rectangle is less than $\pi/8$ and their overlap ratio exceeds 0.5, the estimated rectangle is considered a correct detection. Multiple detections of the same text line are taken as false positives. Thus, the metrics of precision (P'') and recall (R'') can be defined as follows

Table 3 Performance of different text detection methods evaluated on ICDAR datasets

Method	Year	ICDAR2011			ICDAR2013			ICDAR2015			ICDAR2017		
		P(%)	R(%)	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Yao [24]	2016	–	–	–	88.88	80.22	84.33	72.26	58.69	64.77	–	–	–
Zhang [25]	2016	–	–	–	88	78	83	71	43	54	–	–	–
He [33]	2016	88	79	84	90	83	86	–	–	–	–	–	–
Zhong [34]	2016	85	81	83	87	83	85	–	–	–	–	–	–
Gupta [35]	2016	91.5	74.8	82.3	92	75.5	83	–	–	–	–	–	–
Tian [36]	2016	89	79	84	93	83	88	74	52	61	–	–	–
Qin [26]	2017	–	–	–	90	83	86	79	65	71	–	–	–
Dai [27]	2017	–	–	–				88.6	80	84.1	–	–	–
Liao [37]	2017	88	82	**85**	88	83	85	–	–	–	–	–	–
Ma [38]	2017	–	–	–	90	72	80	82.17	73.23	77.44	–	–	–
He [40]	2017	–	–	–	92	81	86	82	80	81	–	–	–
Jiang [41]	2017	–	–	–	93.55	82.59	87.73	85.62	79.68	82.54	–	–	–
Liu [42]	2017	–	–	–	–	–	–	73.23	68.22	70.64	–	–	–
Shi [43]	2017	–	–	–	87.7	83	85.3	73.1	76.8	75	–	–	–
Zhou [44]	2017	–	–	–	–	–	–	83.27	78.33	80.72	–	–	–
He [46]	2017	–	–	–	89	86	88	80	73	77	–	–	–
Deng [28]	2018	–	–	–	88.6	87.5	88.1	85.5	82	83.7	–	–	–
Li [30]	2018	–	–	–	–	–	–	89.3	85.22	87.21	77.01	68.4	**72.45**
Yang [31]	2018	–	–	–	–	–	–	93.8	87.3	**90.5**	–	–	–
Liao [45]	2018	–	–	–	92	86	89	88	80	83.8	–	–	–
Zhong [47]	2018	–	–	–	94	90	92	89	83	86	75	66	70
Lyu [48]	2018	–	–	–	92	84.4	88	89.5	79.7	84.3	74.3	70.6	72.4
He [49]	2018	–	–	–	91	89	90	87	86	87	–	–	–
Liu [69]	2018	–	–	–	–	–	**92.82**	–	–	–	81.86	62.3	70.75
Liao [70]	2018	–	–	–	92	86	89	87.8	78.5	82.9	–	–	–

The significance of bold in the tables means the best result acquired by the method

$$P'' = |TP|/|E| \tag{16}$$

$$R'' = |TP|/|T| \tag{17}$$

where TP is the set of true positive detections, while E and T are respectively the sets of estimated rectangles and ground truth rectangles.

6.2.4 Text Recognition Protocols

Given cropped word image, word recognition accuracy is a commonly used evaluation metric, which is defined as the ratio of the correctly recognized word number to the ground truth number. For holistic scene image containing texts, there are two protocols for evaluation, i.e., word spotting and end-to-end. Word spotting only examines whether the words in lexicon appear in input image, and it ignores symbols, punctuations, numbers and words whose length is less than three. End-to-end protocol concerns both detection and recognition results, and it needs to recognize all the words precisely, no matter whether the lexicon contains these strings. F-measure is also adopted by the two protocols. Performance comparison

Table 4 Performance of different text detection methods evaluated on other public datasets

Method	Year	MSRA TD500			COCO-Text			SVT			RCTW-17		
		P(%)	R(%)	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Yao [24]	2016	76.51	75.31	75.91	43.23	27.1	33.31	–	–	–	–	–	–
Zhang [25]	2016	83	67	74	–	–	–	–	–	–	–	–	–
He [33]	2016	79	65	71	–	–	–	–	–	–	–	–	–
Gupta [35]	2016	–	–	–	–	–	–	26.2	27.4	26.7	–	–	–
Tian [36]	2016	–	–	–	–	–	–	68	65	**66**	–	–	–
Dai [27]	2017	87.6	77.1	82	–	–	–	–	–	–	–	–	–
Ma [38]	2017	82.1	67.7	74.2	–	–	–	–	–	–	–	–	–
He [40]	2017	77	70	74	–	–	–	–	–	–	–	–	–
Shi [43]	2017	86	70	77	–	–	–	–	–	–	–	–	–
Zhou [44]	2017	87.28	67.43	76.08	50.39	32.4	39.45	–	–	–	–	–	–
He [46]	2017	–	–	–	46	31	37	–	–	–	–	–	–
Deng [28]	2018	83	73.2	77.8	–	–	–	–	–	–	–	–	–
Yang [31]	2018	87.5	79	**83**	–	–	–	–	–	–	78.5	56.9	66
Liao [45]	2018	87	73	79	64	57	**61**	–	–	–	77.5	59.1	**67**
Lyu [48]	2018	87.6	76.2	81.5	61.9	32.4	42.5	–	–	–	–	–	–
Liao [70]	2018	–	–	–	60.87	56.7	58.72	–	–	–	–	–	–

The significance of bold in the tables means the best result acquired by the method

Fig. 12 The learned context surrounding the text by deformable PSROI pooling [31]

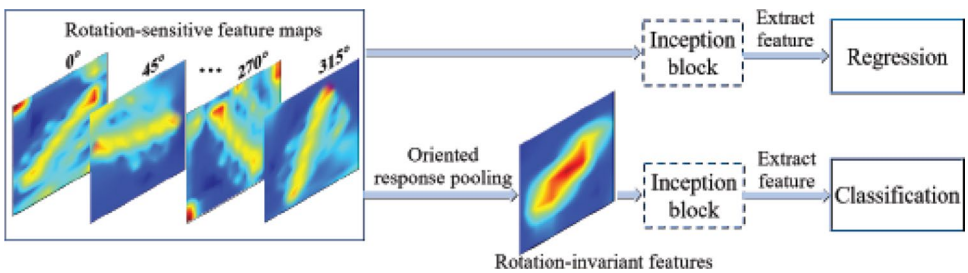


Fig. 13 Rotation sensitive regression [45]

Table 5 Cropped word recognition accuracy (%) on ICDAR datasets

Method	Year	IC03-50	IC03-Full	IC03	IC11-50	IC11-Full	IC13	IC15
Wang [9]	2012	90	84	–	–	–	–	–
Bissacco [51]	2013	–	–	–	–	–	82.83	–
Shi [71]	2013	87.44	79.3	–	87.04	82.87	–	–
Jaderberg [52]	2014	98.7	98.6	–	–	–	90.8	–
Yao [72]	2014	88.48	80.33	–	–	–	–	–
Shi [55]	2015	98.7	97.6	89.4	–	–		–
Jaderberg [57]	2015	97.8	97	89.6	–	–	81.8	–
He [58]	2016	97	93.8		–	–		–
Shi [59]	2016	98.3	96.2	90.1	–	–	88.6	–
Lee [60]	2016	97.9	97	88.7	–	–	90	–
Jaderberg [61]	2016	98.7	98.6	93.3	–	–	90.8	–
Lou [62]	2016	–	–	–	–	–	86.2	–
Yang [65]	2017	–	–	–	–	–	85.21	**79.78**
Bartz [66]	2017	–	–	–	–	–	90.3	
Bai [68]	2018	98.7	97.9	**94.6**	–	–	**94.4**	73.9

The significance of bold in the tables means the best result acquired by the method

Table 6 Cropped word recognition accuracy (%) on other public datasets

Method	Year	SVT-50	SVT	IIIT5 K-50	IIIT5 K-1 k	IIIT5 K
Wang [9]	2012	70	–	–	–	–
Bissacco [51]	2013	90.93	–	–	–	–
Shi [71]	2013	–	73.51	–	–	–
Jaderberg [52]	2014	95.4	80.7	97.1	92.7	–
Almazan [54]	2014	87.01	–	88.57	75.6	–
Yao [72]	2014	–	75.89	80.2	69.3	38.3
Shi [55]	2015	96.4	80.8	97.6	94.4	78.2
Jaderberg [57]	2015	93.2	71.7	95.5	89.6	–
He [58]	2016	93.5	–	94	91.5	–
Shi [59]	2016	95.5	81.9	96.2	93.8	81.9
Lee [60]	2016	96.3	80.7	96.8	94.4	78.4
Jaderberg [61]	2016	95.4	80.7	97.1	92.7	–
Lou [62]	2016	–	80.7	–	–	–
Bartz [66]	2017	–	79.8	–	–	86
Bai [68]	2018	**96.6**	**87.5**	**99.5**	**97.9**	**88.3**

The significance of bold in the tables means the best result acquired by the method

Table 7 End-to-end F-measures (%) on ICDAR03, ICDAR11, ICDAR13 and SVT

Method	Year	IC03-50	IC03-Full	IC03	SVT-50	SVT	IC11	IC13
Wang [9]	2012	72	67	–	46	–	–	–
Jaderberg [61]	2016	90	86	78	76	53	76	76
Gupta [35]	2016	–	–	–	67.7	55.7	84.3	84.7
Gomez [67]	2017	92	90	75	85	54	–	–
Liao [37]	2017	–	–	–	84	64	87	–
Liao [70]	2018	–	–	–	84	64	–	–

In this section, we reported the experimental results of representative text detection and recognition methods on some public datasets through a comprehensive literature review. Since different methods may conduct experiments on different benchmark datasets, and even on the same dataset they may adopt different training sets (such as using synthetic dataset for pre-training, or using special data augmentation scheme to enlarge the number of training samples), it is impossible for us to make an absolutely fair comparison. However, we can witness the development of state-of-the-art methods in this field and acquire some inspiration.

Table 8 Word spotting and end-to-end F-measures (%) on ICDAR13 and ICDAR15

Method	Year	Word Spotting			End-to-end		
		IC13-100	IC13-Full	IC13	IC13-100	IC13-Full	IC13
Gomez [67]	2017	85.37	83.58	70.71	81.16	79.49	68.54
Liao [37]	2017	94	92	87	91	89	84
Liu [69]	2018	95.94	93.9	**87.6**	91.99	90.11	84.77
Liao [70]	2018	**96**	**95**	87	**93**	**92**	**85**
Method	Year	Word Spotting			End-to-end		
		IC15-50	IC15-Full	IC15	IC15-50	IC15-Full	IC15
Gomez [67]	2017	56	52.26	49.73	53.3	49.61	47.18
Liao [37]	2017	-	-	-	-	-	-
Liu [69]	2018	**87.01**	**82.39**	**67.97**	**83.55**	**79.11**	**65.33**
Liao [70]	2018	76.45	69.04	54.37	73.34	65.87	51.9

The significance of bold in the tables means the best result acquired by the method

Tables 3 and 4 report text detection performance of different methods on eight datasets. As mentioned in Sect. 2, deep learning based methods become the mainstream recently for text detection. Here we only give results of this group of methods. As is shown in Table 3, at present the F-measures on ICDAR2013 and ICDAR2015 both exceed 90%. Especially, the performance on ICDAR 2015 has increased drastically from 54% (Zhang et al. [25]) to 90.5% (Yang et al. [31]) in terms of F-measure. In [31], deformable PSROI pooling is applied to add offsets to the spatial binning positions in PSROI pooling (see Fig. 12), which can greatly enhance the performance of multi-oriented text detection. As is shown in Table 4, the F-measures on the other four datasets all achieve unprecedented levels so far. On the largest COCO-Text dataset, the performance has increased drastically from 33.31% (Yao et al. [24]) to 61% (Liao et al. [45]) in terms of F-measure. In [45], a rotation sensitive regression network (see Fig. 13) is adopted, which can be helpful to achieve better detection result. It can be observed that abundant technologies of general object detection and semantic segmentation have been extended for scene text location, and the current trend is applying deep learning framework to training an end-to-end text detector.

Tables 5, 6, 7 and 8 report text recognition performance of different methods on six commonly used datasets. As is shown in Tables 5 and 6, the method of Bai et al. [68] achieve relatively high performance on all ICDAR datasets. In [68], edit probability (EP) is proposed to train attention based text recognition model. By applying a sequence generation mechanism for lexicon-free prediction, this method can effectively recognize out-of-training-set words, and obtain the best result on ICDAR 2003 and ICDAR 2013 without strong or weak lexicon. As is shown in Tables 7 and 8, the methods of Liao et al. [70] and Liu et al. [69] achieve the state-of-the-art performance. Since TextBoxes ++ [70] extends directly from TextBoxes [37] that mainly handles horizontal texts, it obtains relatively high F-measures on ICDAR 2013 and SVT dataset. Note

Fig. 14 Illustration of RoIRotate [69]

that the performance improvement of TextBoxes ++ is spectacularly significant on SVT dataset due to its training on low-resolution images. In [69], the RoIRotate operator is proposed to connect detection and recognition in a unified network, and it can apply transformation on oriented detection bounding boxes to obtain axis-aligned feature maps (see Fig. 14). Therefore, such unified network achieves obvious advantages on oriented ICDAR 2015 dataset. Note that there is no general text recognition method yet, and each method only performs well on certain datasets. As long as the text regions are properly localized, traditional methods have already achieved relatively high cropped word recognition accuracy. However, present methods attempt to construct an end-to-end framework without complicated pre- or post-processing for both text detection and recognition.

7 Conclusions

Scene text detection and recognition have received increasing attention in computer vision due to its potential applications in numerous fields. This paper mainly reviews detection and recognition methods proposed in the last decade. We comprehensively classify these methods and highlight the key techniques. Furthermore, more than 10 benchmark datasets and the corresponding evaluation protocols are described in the paper. Finally, we report the results of more than 40 representative methods and compare their performance. Although great progress has been achieved in text detection and recognition recently, we also find out some problems that should be addressed.

Since most methods focus on text in English, there is still ample room remained for performance improvement on non-Latin or multi-lingual datasets, such as RCTW-17, MSRA-TD500 and ICDAR 2017 MLT. It is potentially to construct a common text detection engine based on character detectors, since character is the most basic element for various languages. Some weakly supervised scene text detection frameworks [106, 107] have been proposed recently, and they can train robust scene text detectors with a small amount of annotated character images. We consider that this work worthy to be further studied in the future. The results on ICDAR 2015 and COCO-Text are also unsatisfactory. It means that we need to tackle the problem of incidental and diversified text detection. Enhancement and rectification methods [22, 31] should be integrated in the conventional deep learning models so as to obtain better performance in the future work. Moreover, many existing text recognition methods achieve poor performance with general lexicons. Schemes of applying large scale language information [108, 109] and sequence leaning [55] have been proposed for text recognition, which should be further studied.

Funding This work is supported in part by the Sub Project of National Key Research and Development Program (2017YFC0804002) and the National Natural Science Foundation of China (61662048, 61772277, 71771125 and 61603192).

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Yin XC, Yin X, Huang K, Hao HW (2014) Robust text detection in natural scene images. IEEE Trans Pattern Anal Mach Intell 36:970–983
2. JJ Weinman, E Learned-Miller, AR Hanson (2009) Scene Text Recognition Using Similarity and a Lexicon with Sparse Belief Propagation. IEEE Transactions on Pattern Analysis & Machine Intelligence 1733-1746
3. Karaoglu S, Tao R, Gevers T, Smeulders AWM (2017) Words matter: scene text for image classification and retrieval. IEEE Trans Multimedia 31:1063–1076
4. Ye Q, Doermann D (2015) Text detection and recognition in imagery: a survey. IEEE Trans Pattern Anal Mach Intell 37:1480–1500
5. Uchida S (2014) Text localization and recognition in images and video. Handbook of document image processing and recognition. Springer, London, pp 843–883
6. Babenko B, Belongie S (2012) End-to-end scene text recognition. In: IEEE international conference on computer vision, pp 1457–1464
7. Pan YF, Hou X, Liu CL (2011) A hybrid approach to detect and localize texts in natural scene images 20:800–813
8. Mishra A, Alahari K, Jawahar CV (2012) Scene text recognition using higher order language priors. In: Proceedings british machine vision conference, pp 1–11
9. Wang T, Wu DJ, Coates A, Ng AY (2012) End-to-end text recognition with convolutional neural networks. In: International conference on pattern recognition, pp 3304–3308
10. Jaderberg M, Vedaldi A, Zisserman A (2014) Deep features for text spotting. In: European conference on computer vision, pp 512–528
11. Epshtein B, Ofek E, Wexler Y (2010) Detecting text in natural scenes with stroke width transform. In: Computer vision & pattern recognition, pp 2963–2970
12. Neumann L, Matas J (2010) A method for text localization and recognition. In: Asian conference on computer vision, pp 770–783
13. L Neumann (2012) Real-time scene text localization and recognition. In: Computer vision & pattern recognition, pp 3538–3545
14. Neumann L, Matas J (2015) Real-time lexicon-free scene text localization and recognition. IEEE Trans Pattern Anal Mach Intell 38:1872–1885
15. Yin XC, Yin X, Huang K, Hao HW (2014) Robust text detection in natural scene images. IEEE Trans Pattern Anal Mach Intell 36:970–983
16. Huang W, Qiao Y, Tang X (2014) Robust scene text detection with convolution neural network induced MSER trees. In: European Conference on Computer Vision, pp 497–511
17. Gomez L, Karatzas D (2015) Object proposals for text extraction in the wild. In: International conference on document analysis and recognition, pp 206–210

18. Buta M, Neumann L, Matas J (2015) FASText efficient unconstrained scene text detector. In: IEEE international conference on computer vision, pp 1206–1214
19. Zhang Z, Shen W, Yao C, Bai X (2015) Symmetry-based text line detection in natural scenes. In: IEEE conference on computer vision and pattern recognition, pp 2558–2567
20. Cho H, Sung M, Jun B (2016) CannyText detector fast and robust scene text localization algorithm. In: IEEE conference on computer vision and pattern recognition, pp 3566–3573
21. Fabrizio J, Robert-Seidowsky M, Dubuisson S, Calarasanu S (2016) TextCatcher: a method to detect curved and challenging text in natural scenes. In: International conference on document analysis and recognition, pp 99–117
22. He T, Huang W, Qiao Y, Yao J (2016) Text-attentional convolutional neural networks for scene text detection. IEEE Trans Image Process 25:2529–2541
23. Zhu Y, Yao C, Bai X (2016) Scene text detection and recognition: recent advances and future trends. Front Comput Sci 10:19–36
24. Yao C, Bai X, Sang N, Zhou X, Zhou S (2016) SceneText detection via holistic, multi-channel prediction. arXiv:1606.09002 pp 1–10
25. Zhang Z, Zhang C, Shen W, Yao C, Liu W (2016) Multi-oriented text detection with fully convolutional networks. In: Computer vision & pattern recognition, pp 4159–4167
26. Qin S, Manduchi R (2017) Cascaded segmentation-detection networks for word-level text spotting. In: International conference on document analysis and recognition, pp 1275–1282
27. Dai Y, Huang Z, Gao Y, Xu Y, Chen K (2017) Fused text segmentation networks for multi-oriented scene text detection, pp 1–6. arXiv:1709.03272
28. Deng D, Liu H, Li X, Cai D (2018) PixelLink: detecting scene text via instance segmentation. In: Proceedings of association for the advancement of artificial intelligence, pp 1–8
29. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S (2016) SSD: single shot multibox detector. In: European conference on computer vision, pp 21–37
30. Li X, Wang W, Hou W, Liu RZ, Lu T (2018) Shape robust text detection with progressive scale expansion network, pp 1–12. arXiv:1806.02559
31. Yang Q, Cheng M, Zhou W, Chen Y, Qiu M (2018) IncepText: a new inception-text module with deformable psroi pooling for multi-oriented scene text detection. In: International joint conference on artificial intelligence, pp 1–7
32. Dai J, Qi H, Xiong Y, Li Y, Zhang G (2017) Deformable convolutional networks. In: IEEE international conference on computer vision, pp 764–773
33. He T, Huang W, Qiao Y, Yao J (2016) Accurate text localization in natural image with cascaded convolutional textnetwork, pp 1–10. arXiv:1603.09423
34. Zhong Z, Jin L, Zhang S, Feng Z (2016) DeepText a unified framework for text proposal generation and text detectionin natural images, pp 1–12. arXiv:1605.07314v1
35. Gupta A, Vedaldi A, Zisserman A (2016) Synthetic data for text localisation in natural images. In: IEEE conference on computer vision and pattern recognition, pp 2315–2324
36. Tian Z, Huang W, He T, He P, Qiao Y (2016) Detecting text in natural image with connectionist text proposal network. In: European conference on computer vision, pp 56–72
37. Liao M, Shi B, Bai X, Wang X, Liu W (2017) TextBoxes a fast text detector with a single deep neural network. In: Proceedings of association for the advancement of artificial intelligence, pp 1–7
38. Ma J, Shao W, Ye H, Wang L, Wang H (2017) Arbitrary-oriented scene text detection via rotation proposals. IEEE Trans Multimed 20:1–9
39. Girshick R (2015) Fast R-CNN. In: IEEE international conference on computer vision, pp 1440–1448
40. He W, Zhang XY, Yin F, Liu CL (2017) Deep direct regression for multi-oriented scene text detection. In: IEEE international conference on computer vision, pp 745–753
41. Jiang Y, Zhu X, Wang X, Yang S, Li W (2017) R2CNN: rotational region cnn for orientation robust scene text detection, pp 1–8. arXiv:1706.09579
42. Liu Y, Jin L (2017) Deep matching prior network toward tighter multi-oriented text detection. In: IEEE conference on computer vision and pattern recognition, pp 3454–3461
43. Shi B, Bai X, Belongie S (2017) Detecting oriented text in natural images by linking segments. In: IEEE conference on computer vision and pattern recognition, pp 2482–3490
44. Zhou X, Yao C, Wen H, Wang Y, Zhou S (2017) EAST an efficient and accurate scene text detector. In: IEEE conference on computer vision and pattern recognition, pp 2642–2651
45. Liao M, Zhu Z, Shi B, Xia G, Bai X (2018) Rotation-sensitive regression for oriented scene text detection. In: IEEE conference on computer vision and pattern recognition, pp 1–10
46. He P, Huang W, He T, Zhu Q, Qiao Y (2017) Single shot text detector with regional attention. In: IEEE international conference on computer vision, pp 3047–3055
47. Zhong Z, Sun L, Huo Q (2018) An Anchor-Free Region proposal network for faster R-CNN based text detection approaches, pp 1–8. arXiv:1804.09003
48. Lyu P, Yao C, Wu W, Yan S, Bai X (2018) Multi-oriented scene text detection via corner localization and region segmentation. In: IEEE conference on computer vision and pattern recognition, pp 1–10
49. He T, Tian Z, Huang W, Shen C, Qiao Y (2018) Single shot text spotter with explicit alignment and attention. In: ieee conference on computer vision and pattern recognition, pp 1–10
50. He K, Gkioxari G, Dollar P, Girshick R (2018) Mask R-CNN. IEEE transactions on pattern analysis & machine intelligence
51. Bissacco A, Cummins M, Netzer Y, Neven H (2013) Photo OCR: reading text in uncontrolled conditions. In: IEEE international conference on computer vision, pp 785–792
52. Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2014) Synthetic data and artificial neural networks for natural scene text recognition. In: Conference on neural information processing systems, pp 1–10
53. Yao C, Bai X, Liu W (2014) A unified framework for multi-oriented text detection and recognition. IEEE Trans Image Process 23:4737–4749
54. Almazan J, Gordo A, Fornes A, Valveny E (2015) Word spotting and recognition with embedded attributes. IEEE Trans Pattern Anal Mach Intell 36:2552–2566
55. Shi B, Bai X, Yao C (2016) An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans Pattern Anal Mach Intell 39:2298–2304
56. Shi B, C Yao, C Zhang, X Guo (2015) Automatic script identification in the wild. In: International conference on document analysis and recognition, pp 531–535
57. Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2015) Deep structured output learning for unconstrained text recognition. In: International conference on learning representations, pp 1–10
58. He P, Huang W, Qiao Y, Loy CC, Tang X (2016) Reading scene text in deep convolutional sequences. In: Proceedings of association for the advancement of artificial intelligence, pp 1–8
59. Shi B, Wang X, Lyu P, Yao C, Bai X (2016) Robust scene text recognition with automatic rectification. In: IEEE conference on computer vision and pattern recognition, pp 1–9
60. Lee CY, Osindero S (2016) Recursive recurrent nets with attention modeling for OCR in the Wild. In: IEEE conference on computer vision and pattern recognition, pp 2231–2239

61. Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2016) Reading text in the wild with convolutional neural networks. Int J Comput Vis 116:1–20
62. Lou X, Kansky K, Lehrach W, Laan V, Marthi B (2016) Generative shape models joint text recognition and segmentation with very little training data. In: Advances in Neural Information Processing Systems. Neural Information Processing Systems Foundation, Barcelona
63. Kang C, Kim G, Yoo SI (2017) Detection and recognition of text embedded in online images via neural context models. In: Proceedings of association for the advancement of artificial intelligence, pp 4103–4110
64. B Moysset, C Kermorvant, C Wolf (2017) Full-Page Text Recognition Learning Where to Start and When to Stop. In: International Conference on Document Analysis and Recognition, pp 871-876
65. Yang C, Yin XC, Li Z, Wu J, Guo C(2017) AdaDNNs: adaptive ensemble of deep neural networks for scene text recognition, pp 1–8. arXiv:1710.03425
66. Bartz C, Yang H, Meinel C (2017) STN-OCR a single neural network for text detection and text recognition, pp 1-9. arXiv :1707.08831
67. Gomezbigorda L, Karatzas D (2017) TextProposals a text-specific selective search algorithm for word spotting in the wild. Pattern Recognit 70:60–74
68. Bai F, Cheng Z, Niu Y, Pu S, Zhou S (2018) Edit probability for scene text recognition, pp 1–9. arXiv:1805.03384
69. Liu X, Liang D, Yan S, Chen D, Qiao Y (2018) FOTS Fast oriented text spotting with a unified network, pp 1-10. arXiv:1801.01671
70. Liao M, Shi B, Bai X (2018) TextBoxes ++ a single-shot oriented scene text detector. IEEE Trans Image Process 27:3676–3690
71. Shi C, Wang C, Xiao B, Zhang Y, Gao S (2013) Scene text recognition using part-based tree-structured character detection. In: IEEE conference on computer vision and pattern recognition, pp 2961–2968
72. Bai X, Yao C, Liu W (2014) Strokelets a learned multi-scale representation for scene text recognition. In: IEEE conference on computer vision and pattern recognition, pp 4042–4049
73. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: IEEE conference on computer vision and pattern recognition, pp 3431–3440
74. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition, pp 1–12. arXiv:1512.03385
75. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: IEEE conference on computer vision and pattern recognition, pp 779–788
76. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: International Conference on Neural Information Processing Systems, pp 91–99
77. Xie S, Tu Z (2015) Holistically-nested edge detection. In: International Journal of Computer Vision, pp 1–16
78. Postl W (1986) Detection of linear oblique structures and skew scan in digitized documents. In: International Conference on Pattern Recognition, pp 687-689
79. Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pp 379–387
80. Zhou Y, Ye Q, Qiu Q, Jiao J (2017) Oriented response networks. In: IEEE conference on computer vision and pattern recognition, pp 4961–4970
81. Yao C, Zhang X, Bai X, Liu W, Ma Y, Tu Z (2012) Detecting texts of arbitrary orientations in natural images. In: IEEE international conference on computer vision, pp 1083–1090
82. Karatzas D, Antonacopoulos A (2004) Text extraction from web images based on a split-and-merge segmentation method using colour perception. In: International conference on pattern recognition, pp 634–637
83. Rajendran D, Shivakumara P, Su B, Lu S, Tan CL (2011) A new Fourier-moments based video word and character extraction method for recognition. In: International conference on document analysis and recognition, pp 1165–1169
84. Sharma N, Shivakumara P, Pal U, Blumenstein M, Tan CL (2012) A new method for arbitrarily-oriented text detection in video. In: Proceedings of the IAPR international workshop on document analysis systems, pp 74–78
85. Shivakumara P, Sreedhar R, Phan T, Lu S, Tan CL (2012) Multioriented video scene text detection through Bayesian classification and boundary growing. IEEE Trans Circuits Syst Video Technol 22:1227–1235
86. Singh C, Bhatia N, Kaur A (2008) Hough transform based fast skew detection and accurate skew correction methods. Pattern Recognit 41:3528–3546
87. Yi C, Tian Y (2011) Text string detection from natural scenes by structure-based partition and grouping. IEEE Trans Image Process 20:2594–2605
88. Shivakumara P, Phan TQ, Tan CL (2011) A Laplacian approach to multi-oriented text detection in video. IEEE Trans Pattern Anal Mach Intell 33:412–419
89. Pan YF, Hou X, Liu CL (2011) A hybrid approach to detect and localize texts in natural scene images. IEEE Trans Image Process 20:800–813
90. Alsharif O, Pineau J (2013) End-to-end text recognition with hybrid HMM Maxout models. sys, pp 1–10. arXiv:1310.1811v1
91. Jawahar CV, Alahari K, Mishra A (2012) Top-down and bottom-up cues for scene text recognition. In: IEEE conference on computer vision and pattern recognition, pp 2687–2694
92. Novikova T, Barinova O, Kohli P, Lempitsky V (2012) Large-lexicon attribute-consistent text recognition in natural images. In: European conference on computer vision, pp 752–765
93. Graves A, Gomez F (2006) Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: International conference on machine learning, pp 369–376
94. http://www.iapr-tc11.org/mediawiki/index.php?title=ICDAR_2003_Robust_Reading_competitions. Accessed July 2018
95. http://www.cvc.uab.es/icdar2011competition/?com=downloads. Accessed July 2018
96. http://rrc.cvc.uab.es/?ch=2&com=downloads. Accessed July 2018
97. http://rrc.cvc.uab.es/?ch=4&com=downloads. Accessed July 2018
98. http://rrc.cvc.uab.es/?ch=8&com=introduction. Accessed July 2018
99. http://www.iapr-tc11.org/mediawiki/index.php/MSRA_Text_Detection_500_Database_(MSRA- TD500). Accessed July 2018
100. https://vision.cornell.edu/se3/coco-text-2/. Accessed July 2018
101. http://vision.ucsd.edu/~kai/grocr/. Accessed July 2018
102. http://rctw.vlrlab.net/. Accessed July 2018
103. http://cvit.iiit.ac.in/research/projects/cvit-projects/the-iiit-5k-word-dataset. Accessed July 2018
104. http://www.robots.ox.ac.uk/~vgg/data/scenetext/. Accessed July 2018
105. http://www.robots.ox.ac.uk/~vgg/data/text/. Accessed July 2018
106. Tian S, Lu S, Li C (2017) WeText scene text detection under weak supervision. In: IEEE international conference on computer vision, pp 1501–1509
107. Hu H, Zhang C, Luo Y, Wang Y, Han J(2017) WordSup: exploiting word annotations for character based text detection. In: IEEE international conference on computer vision, pp 4950–4959

108. Weinman JJ, Butler Z, Knoll D, Field J (2014) Toward integrated scene text reading. IEEE Trans Pattern Anal Mach Intell 36:375–387

109. Bai X, Yao C, Liu W (2016) Strokelets: a learned multi-scale mid-level representation for scene text recognition. IEEE Trans Image Process 25:2789–2802