Received: 8 March 2022 | Revised: 13 May 2022 | Accepted: 11 June 2022

IET Image Processing

DOI: 10.1049/ipr2.12574

The Institution of Engineering and Technology WILEY

REVIEW

A survey on methods, datasets and implementations for scene text spotting

Pablo Blanco-Medina[1,2] | Eduardo Fidalgo[1,2] | Enrique Alegre[1,2] | Víctor González-Castro[1,2]

[1] Department of Electrical, Systems and Automatic Engineering, Universidad de León, León, Spain

[2] Researcher at INCIBE (Spanish National Institute of Cybersecurity), León, Spain

Correspondence
Pablo Blanco-Medina, Department of Electrical, Systems and Automatic Engineering. Universidad de León, León, Spain.
Email: pablo.blanco@unileon.es

Funding information
Instituto Nacional de Ciberseguridad, Grant/Award Numbers: Addendum 22, Addendum 01; Universidad de León, Grant/Award Number: Ayudas Estudios Doctorado 2018

Abstract

Text Spotting is the union of the tasks of detection and transcription of the text that is present in images. Due to the various problems often found when retrieving text, such as orientation, aspect ratio, vertical text or multiple languages in the same image, this can be a challenging task. In this paper, the most recent methods and publications in this field are analysed and compared. Apart from presenting features already seen in other surveys, such as their architectures and performance on different datasets, novel perspectives for comparison are also included, such as the hardware, software, backbone architectures, main problems to solve, or programming languages of the algorithms. The review highlights information often omitted in other studies, providing a better understanding of the current state of research in Text Spotting, from 2016 to 2022, current problems and future trends, as well as establishing a baseline for future methods development, comparison of results and serving as guideline for choosing the most appropriate method to solve a particular problem.

1 | INTRODUCTION

Text Spotting consists of the extraction of text present in visual media by means of the detection of regions and their successive transcription. First, in the detection phase, areas containing text are delimited. Thereafter, these regions are processed in the recognition phase to achieve the final transcription for each detected area. Text Spotting can be applied to obtain information on real scenes, such as a traffic sign or advertising panels [1]. It is also possible to extract artificially added text to an image, such as subtitles or watermarks [2]. Applications of Text Spotting can be found in the Industrial field, such as assembly lines [3, 4], video indexing [1], document analysis [5], robot navigation [6, 7], automatic classification of Information / Operational Technology (IT/OT) snapshots in Industrial Control Systems [8], or identification of port containers [9]. In CyberSecurity, it can be applied to retrieve text found in images from Tor (The Onion Router) darknet, which can be linked to the sale of weapons, document falsification [10] from suspicious domains [11] or from child sexual abuse (CSA) images.

Literature presents surveys on Text Spotting [1, 2, 12], which compare proposals and show the progress and recent advances of each task through the use of Convolutional Neural Networks (CNN) [13]. These studies highlight the issues that can make text extraction a difficult task, such as slanted or oriented text displayed with low resolution, in multiple languages, partially occluded, oversaturated, or in different fonts and text sizes [10]. Figure 1 illustrates some examples of challenging scene text.

The proposed solutions may differ in their scope, implementation, datasets for training and testing, programming language, or multiple method configurations [13, 14]. Due to these problems, properly comparing the different methods can be an arduous task [15]. Moreover, searching for a solution that could be integrated into a real tool or service [16, 17] with specific requirements in particular environments (e.g., forensic software applications) may represent a challenge for researchers [18].

Previous state-of-the-art studies often miss these details [1, 2], for example, the hardware and software (HW / SW), the programming language, or the dependencies required by text spotting methods. When working in industrial projects that

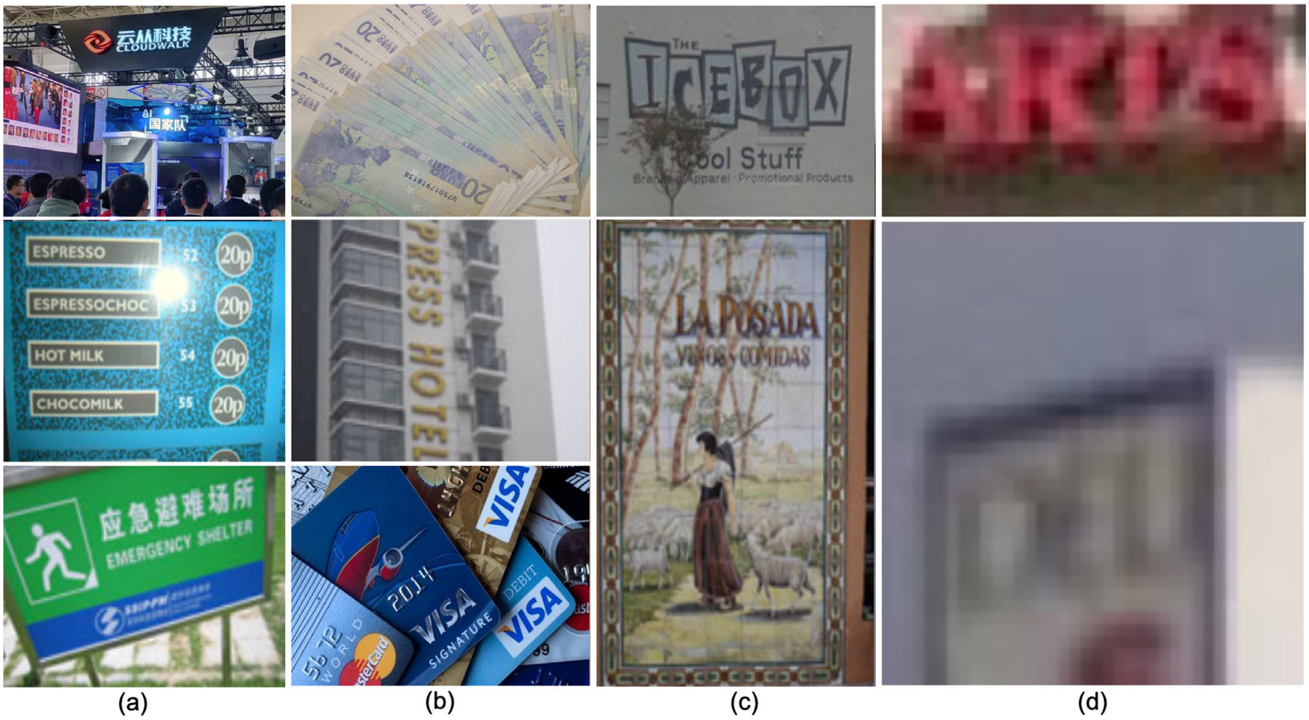

FIGURE 1 Examples of real-scene challenging text in both detection and recognition, taken from state-of-the-art datasets, organised in columns. Multi-language text (a), text in non-horizontal orientations (b), partially occluded text (c), and low-resolution text (d)

require real-time performance, these aspects are essential for effective and successful application of any method [19]. Due to the relevance of aspects such as machine time or production costs in industrial applications, it is necessary to optimise the HW/SW to carry out a particular task. Combined with the performance comparison in terms of precision, recall or F-1 score, this information would facilitate the selection of the most appropriate algorithm for a Text Spotting oriented task, resulting in a more effective integration of recent research advances in real projects.

To the best of our knowledge, there are no studies of Text Spotting that cover all the above-mentioned details. For this reason, this article aims to provide a detailed review of recent detection and recognition methods from a wider perspective. Our objective is to assist researchers looking to integrate Text Spotting into industrial applications, as well as defining a procedure for selecting the most suitable algorithm for their research and future scientific contribution.

In this survey, we compare 40 scene-text based proposals published between 2016 and 2022, including approaches not covered in previous studies [13, 15, 20] contributing with a novel viewpoint, where we focus on problem-specific techniques, HW/SW information, programming language and deep learning architectures used by the models. Our analysis differs from other state-of-the-art reviews by highlighting the implementation details, hardware and software specifications of recent methods that achieve cutting-edge results in the detection and recognition tasks. Following our analysis, we recommend the best-suited methods for each task, analysing the most relevant datasets and architectures for the various environments text spotting can be applied to. We also present the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology for our review, which can be seen in Figure 2. At the end of the paper selection process, we had 40 methods for our analysis that

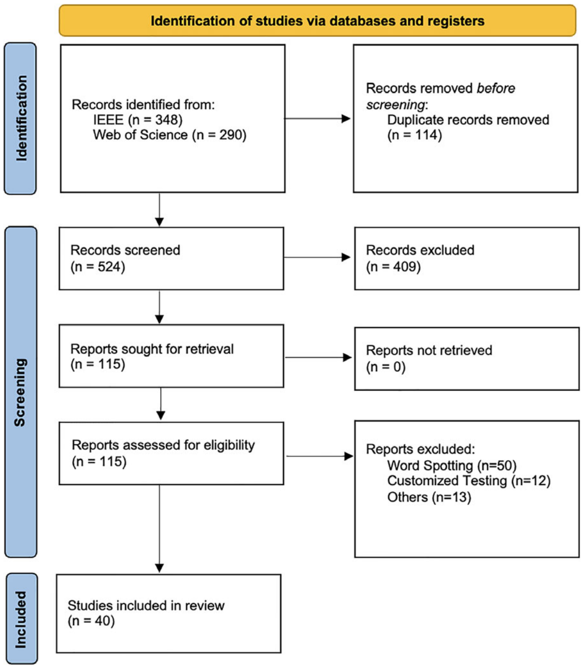

FIGURE 2 PRISMA summary

cover detection, recognition, and end-to-end scene text retrieval systems.

The structure of this article is divided as follows. Section 2 summarizes the previous studies of state of the art on Text Spotting and its applications in industry. Section 3 presents the review methods used for article selection and the PRISMA diagram as a visual summary. Section 4 describes the most widely used datasets in text detection and recognition tasks. Section 5 presents the most common methodologies and tasks in Text Spotting. Next, Section 6 discusses the architectures employed, together with the software and hardware implementations. Section 7 briefly summarizes the problems gathered from our study, how they are correlated, and why authors should focus on improving them to enhance the results of their methods. Lastly, we end the article with the conclusions drawn from our review in Section 8.

2 | LATEST SURVEYS ON TEXT SPOTTING

Ye et al. [1] conducted a study of 10 text detection methods and eight text recognition methods in images, measuring their performance in the SVT (Street View Dataset) [21] and various ICDAR (International Conference on Document Analysis and Recognition) datasets [22]. They also reflected the problem of incidental text, which consists of text that is not directly focused, complicating its detection. Their analysis did not compare the computational costs of the methods, the software and hardware used, or the different possible configurations of the methods presented.

The authors highlighted the issues of multilingual text and real-time performance of current systems [23]. Other recent surveys have analysed specific language-based techniques such as Urdu [24], but also omit information related to their implementation or their multiple configurations.

Yin et al. [2] evaluated Text Spotting in both images and video, grouping and classifying methods, datasets, main problems and future task challenges, but omitting direct comparisons between methods.

Elaalyani et al. [25] reviewed a total of 17 methods, analysing their results in the ICDAR 2011 dataset. Their work did not detail the main problems to be solved by each method or their limitations. Furthermore, due to the age of the dataset, this study omitted an analysis of more recent problems, such as partial occlusion or multilingual text, which hinders a fair comparison of the most recent approaches.

Long et al. [12] compared 28 detection and 13 recognition methods on five and eight datasets, respectively. The authors highlighted three main problems: diversity and variability of the text, complexity of the backgrounds and uncontrolled text situations, such as low resolution or blurred content. This study analysed the use of auxiliary techniques such as synthetic image generation [26] or techniques that allow blurry regions to be corrected. The authors also detailed the use of contextual information to ignore regions with a lower probability of containing text, such as faces. However, the problem of text orientation was not addressed, nor was the analysis of rectification techniques, which allows correcting the orientation of text in an image by facilitating its transcription [13, 14, 27].

Baek et al. [15] analysed the recognition task by comparing 12 methods and generating their own text recognition structure divided into four phases: transformation, feature extraction, sequential modelling and prediction. Using multiple configurations for each phase, the authors reflected the performance and computational cost of various method combinations, as well as network architectures. This study highlighted the problems of current datasets, such as incorrect region labelling. They analysed the performance of the Thin Plate Spline (TPS), a variant of spatial transformation networks [28], and the improved performance of ResNet-based architectures over those based on VGG. However, some of the more recent rectification approaches [13, 14] are omitted. The use of these techniques would be relevant in the context of the first phase, as the authors only propose the application of Thin Plate Spline (TPS).

Lin et al. [23] compared 25 detection methods and 16 recognition methods over 9 datasets according to multiple criteria and evaluation subsets. In addition, they surveyed six end-to-end methods, where both detection and recognition were trained together. In this study, the authors explored the architectures that highlight the use of Fully Connected Networks (FCN) for the segmentation task, ResNet [29] and Fast Region-based Convolutional Neural Networks (R-CNN), as well as how different loss functions affect training and evaluation phases. Despite this, the methods were not compared at the computational cost level or the libraries used.

Liu et al. [5] presented a total of 41 methods evaluated on 9 datasets, divided into 22 detection, 15 recognition, and eight end-to-end methods. The analysis compared the computational time of the methods, identifying the problem of their efficiency, by reducing their execution time on computers without GPU support. The problems of detection robustness and multilanguage text also stood out. Despite the number of methods analysed, the study omitted those focused on oriented text correction [13, 20, 27].

Chen et al. [30] focused on the issues of the recognition task, highlighting recent advances, differences with Optical Character Recognition (OCR) approaches and auxiliary techniques such as super-resolution, rectification, and background removal. They compared a total of 66 recognizers against eight datasets. Although they highlighted the lack of a uniform evaluation protocol that addresses multiple configurations and specific problems, they did not include the low-level implementation details of the methods.

Lastly, Gupta et al. [31] studied advances in text spotting tasks, highlighting their recent progress, as well as transfer learning-based methods, studying their performance in the most relevant datasets. Despite this, they also omit some of the finer details of the methods, which can help to select the best architectures and implementations for real-time applications.

Although all of these studies have made comparisons between state-of-the-art methods, they have omitted the inclusion of low-level details regarding the implementation of these approaches. The creation of a unified evaluation framework, with problem-specific datasets, performance protocols and uniform testing procedures would establish a strong baseline to improve both components of Text Spotting further.

3 | PAPER SELECTION

We searched the literature from January 1, 2016, to January 1, 2022, in two databases: (1) IEEE Xplore Digital Library; and (2) Web of Science. We focused on the results obtained from queries of the keywords "Text Spotting", "Scene Text", "Text Detection", and "Text Recognition". We filtered the results by computer science, engineering, and image processing related journals, conferences, and magazines.

From this search, we obtained 638 articles, of which we removed 114 that were duplicated. Next, we studied the titles and abstracts of the remaining 524 nonduplicated papers. After analysing their title, abstract and keywords, we removed studies related to video tasks, handwritten documents and surveys and competition reports, as well as tasks not related to text spotting such as text classification, handwritten text, document analysis using OCR methods, named entity recognition, or sentiment analysis.

The remaining 115 articles were assessed after retrieving the full document. We removed articles related to Word Spotting, which focuses on handwritten documents, as well as articles that did not include state-of-the-art dataset results and articles that presented only theoretical contributions. Our analysis and criteria can be seen summarized in the PRISMA diagram presented in Figure 2.

In the end, we have considered 40 papers for our analysis. From these, we retrieved data regarding the following details: (1) problem focus, (2) software and hardware choice of implementation, (3) techniques implemented, (4) datasets used for training and testing, (5) architectures implemented, and (6) speed-performance comparison. The funders had no role in the procedure of this review.

4 | DATASETS AND PERFORMANCE

In order to measure the performance of both text detection and recognition, methods are tested using a variety of datasets that contain regular and irregular text, that is, text that is not horizontal and frontal [1, 12, 15].

Due to the difficulties found in the labelling task for scene text detection and recognition, most methods use synthetically generated datasets such as MJSynth [37] and SynthText [26] as a first approach to training their proposals. Although MJSynth was designed exclusively for recognition, SynthText is suitable for both tasks. These datasets are generated by placing random words on images in an automated way. The resulting images allow verifying the robustness of the methods against words or phrases located in arbitrary locations. They also contain a larger amount of images when compared to other state-of-the-art datasets, but the cropping of training sets is not a shared process across methods [20]. Lastly, methods can also choose to train with privately-stored data [45].

Regarding non-synthetically generated datasets, the labelling procedure consists of registering the location of the areas of each image where the text is located, as well as the transcription of the same. This process does not have to include all the text of an image, being able to detail only those regions that are of particular interest. The case of ICDAR 2013 [35] stands out, which introduces the concept of *Don't Care* regions, which omit text considered not relevant. Depending on the level of detail of the labelling, information such as text readability, language, or orientation can be included. These details allow for filtering prior to performance evaluation [39]. Furthermore, each individual dataset can have its own evaluation method [46].

In this study, we reviewed 17 datasets, containing both regular and irregular types of text. We focused on the most commonly-used datasets for method evaluation, excluding recent publications that have not been included in most performance analysis methods [47, 48] and datasets that focus on artistic images, as their use is sporadic in current state-of-the-art approaches [23]. Three of these datasets focus solely on the detection task, while four focus on the recognition problem. The remaining nine include information for the evaluation of both tasks, making them suitable for end-to-end approaches.

While most datasets mainly focus on the English language, due to its widespread use and the similarities with other Latin alphabets. ICDAR 2017 MLT (Multi-Lingual Text) [40], seeks to add greater diversity in the languages present in the image that pose additional challenges, due to the differences in character location and transcription between word sequences. Additionally, this dataset corrects problems found in previous labels, and updates the images to include newer issues, such as oriented images.

The latest revision of the ICDAR MLT dataset in 2019 included more images and a total of 10 different languages, and an additional synthetic version of more than 277,000 images, which focus on the task of end-to-end text spotting.

Table 1 shows the information collected about these datasets, retrieved from their original publications or sources.

4.1 | Performance

Measuring the performance of the Text Spotting task is difficult, due to several factors, for example, differences in proposed techniques, method configurations, architectures implemented, model size and computational cost, parameter number, images used for training, or whether CPU or GPU was used.

Although there is no unified protocol to present results, these are commonly measured in terms of precision, recall, and F-1 score in detection, while only precision is given for most recognition approaches in conjunction with the use of lexicons [15, 30]. For end-to-end systems, computational cost per image is also a common metric [49], despite the fact that not all methods use the same dimensions.

TABLE 1 Main features of the analysed datasets, including their number of images for training and testing, the year of publication, the task for which the dataset is used, and the languages of the text contained. 'Multiple' summarises several languages

Dataset	Test	Train	Year	Task	Languages	Image Type	Text Type
ICDAR 2011 [32]	225	229	2011	Detection and recognition	English	Real	Regular
SVT [21]	250	100	2011	Detection and recognition	English	Real	Regular
MSRA-TD500 [33]	200	300	2012	Detection	English and Chinese	Real	Multiple
IIIT5K-WORDS [34]	3,000	2,000	2012	Recognition	English	Real	Regular
ICDAR 2013 [35]	233	229	2013	Detection and recognition	English	Real	Regular
SVT-P [36]	639	-	2013	Recognition	English	Real	Irregular
MJSynth [37]	-	9,000,000	2014	Recognition	English	Synthetic	Multiple
CUTE-80 [38]	288	-	2014	Recognition	English	Real	Irregular
ICDAR 2015 [22]	500	1,000	2015	Detection and recognition	English	Real	Irregular
SynthText [26]	-	800,000	2016	Detection and recognition	English	Synthetic	Multiple
COCO-Text [39]	20,000	43,686	2016	Detection and recognition	English	Real	Multiple
ICDAR 2017 MLT [40]	9,000	9,000	2017	Detection and recognition	Multiple	Real	Multiple
RCTW-17 [41]	4,229	8,034	2017	Detection and recognition	Chinese	Real	Multiple
TotalText [42]	300	1,225	2017	Detection and recognition	English	Real	Multiple
FORU	1,219	3,874	2017	Detection	English and Chinese	Real	Multiple
Multilingual [43]	239	248	2017	Detection	English and Chinese	Real	Multiple
ICDAR 2019 MLT [44]	10,000	10,000	2019	Detection and recognition	Multiple	Real	Multiple

TABLE 2 Most common datasets in the surveyed methods

Dataset	Detection	Recognition	End-to-end
ICDAR 2011 [32]	2	0	2
SVT [21]	3	15	3
MSRA-TD500 [33]	7	0	1
IIIT5K-WORDS [34]	0	**16**	1
ICDAR 2013 [35]	10	15	**5**
SVT-P [36]	0	15	0
MJSynth [37]	0	8	0
CUTE-80 [38]	1	15	0
ICDAR 2015 [22]	**14**	13	3
SynthText [26]	9	14	**5**
COCO-Text [39]	4	1	3
ICDAR 2017 MLT [40]	2	0	2
RCTW-17 [41]	1	0	1
TotalText [42]	4	0	2

Alongside their focus on particular problems, such as curved or oriented text, or on computational cost, these differences highlight the difficulties of making fair comparisons across state-of-the-art algorithms. Table 2 highlights the datasets used in the methods surveyed. For recognition, IIIT5K-Words was the most popular, but closely followed by SVT, ICDAR 2013 and CUTE-80. For end-to-end methods, ICDAR 2013 and SynthText tied for first place.

Despite their improvements over previous datasets, recent datasets are still not widespread, in favour of older versions that are easier to evaluate.

4.1.1 | Text detection

The performance of the detection step is measured by comparing the detected text areas with those documented as 'ground truths', which indicate the regions of interest to be detected by the algorithms. Figure 3 presents the bounding boxes resulting from the application of state-of-the-art text detectors to images that contain difficulties such as multiple fonts and complex backgrounds.

To associate detected areas to documented ones, Intersection over Union (IoU) is used. This metric indicates a percentage of the minimum area that both regions must share to consider the detection correct. To obtain the best possible results, high percentages of IoU over 80% are used, resulting in greater reliability [1].

Specialised tools such as DetEval software [46] can be used to evaluate detection. This tool analyses the performance of an algorithm in the detection phase, by verifying that the detected zones share a minimum area with any of the labelled regions. DetEval also allows for filtering possible overlays with other zones, in addition to checking that all labelled regions have an associated region in the detection results. These comparison methods cannot be used in more complex detections that must be delimited by polygonal structures due to their skew. In such cases, verification should be done comparing these areas according to other criteria to increase reliability [52].

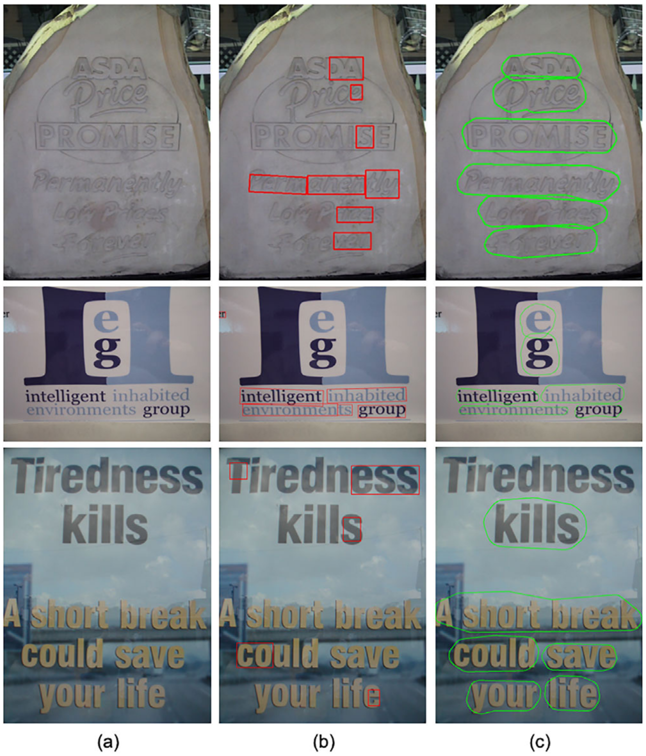

FIGURE 3 Visual performance evaluation of text detectors on images with multiple fonts, different backgrounds and hard-to-identify text. Original images (a), and results from CRAFT [50] (b) and DB [51] (c)

As an alternative performance measurement, Yao et al. [33] proposed an evaluation protocol that includes not only the overlap ratio between the rectangles but also the angle between them. According to their criteria, multiple detections of the same text region are considered false positives, resulting in a stricter evaluation.

4.1.2 | Text recognition

Recognition performance is measured by comparing the result of the transcription with the ground truth text documented in the dataset. This task is more challenging than OCR due to complex backgrounds [53], multiple fonts and poor image quality among other factors [23]. Figure 4 presents cropped regions of the issues that recent text recognizers have focused on [13, 14].

There are two main evaluation criteria for recognition tasks, known as Word Spotting and End-to-End [14]. Word Spotting consists of comparing the transcribed strings against their documented ground truths, while end-to-end refers to the correct transcription of previously extracted areas by text detection, evaluating both tasks at once. Words of three or fewer letters are usually ignored in the evaluation of the methods [22].

To further improve recognition performance, structures known as lexicons or dictionaries can be used in their evaluation, although their usage is optional [14, 54, 55]. They contain word sequences against which transcription is compared, using string matching techniques [56]. Depending on the evaluation criteria, the closest word in the dictionary can be chosen as a valid transcription or as a rejection if it is not a complete match. The most widely used distance measure is the Levenshtein distance [56].

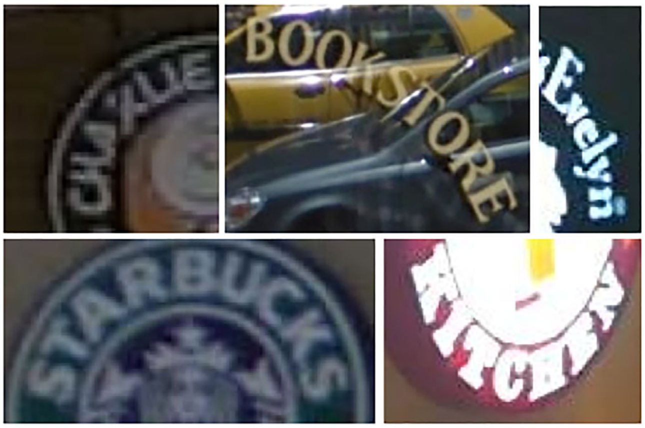

FIGURE 4 Multi-orientation and resolution-based text regions, a recurring issue in modern text transcription

Dictionaries can be classified according to their degree of contextualisation as strong, weak or intermediate, which consists of containing the exact words that appear in the images. Although the use of these methods improves the performance of the task, it also entails an increase in computational cost related to the size of the dictionary and the search algorithm used [57]. Typically, the results obtained are parsed by associating the dictionaries of 50 and 1,000 words [12], which are usually provided by the evaluation datasets [34].

When using dictionaries, the total edit distance can also be reported as a performance measure, although it is not commonly used across most algorithms. This metric compares the labelled string with the transcription, giving a numerical value to the distance between both words for each of the images within the test set [17, 22, 35].

4.2 | Text detection datasets

Due to the multiple problems found in text detection such as multilingual, variable size, oriented or partially occluded text [1], there are multiple datasets focused specifically on the text detection task.

For text in multiple languages, it is worth mentioning the FORU dataset, crawled from images crawled from the Flickr social network. The Multilingual dataset [43] is also relevant, containing text from different languages such as Arabic, Bangla, Chinese, Devanagari, English, French, German, Italian, Japanese, and Korean. In the case of the Chinese language, the RCTW-17 dataset [41], used in the official ICDAR 2017 Robust Reading for Chinese Language competition, stands out.

Among the datasets focused on oriented text, MSRA-TD500 [33] stands out as one of the first datasets designed to address the problem of locating vertical text. One of the most recent datasets, TotalText [42], offers an alternative to this type of image, focussing on curved text labelling that reflects a current problem with text detectors and with a larger number of training images.

4.3 | End-to-end datasets

In the recognition task, the text transcription of all relevant regions of the image is performed. The most used dataset for text recognition is IIIT5K-WORDS [34], generated from real images downloaded from Google Image search. The text found within these images is presented in a horizontal distribution, which is known as regular text.

Multiple datasets include information from both the text transcription and the regions present in an image, applied in methods that seek to solve both problems. The dataset presented by the biennial ICDAR competition is the most referenced in the execution of these end-to-end systems. Its earliest versions, ICDAR 2011 [32] and 2013 [35] are still used to evaluate the most recent proposals [23].

The SVT dataset [21] is another of the most referenced datasets, both in detection and in recognition. It contains multiple languages and is mainly focused on a clear display of names in buildings. However, it also displays low-quality images that include low-resolution and blurred text.

Images can also present text that is curved, distorted, or rotated, increasing the difficulty of the recognition task. This is the case with the SVT-Perspective dataset [36], which contains images taken from various angles that complicate character reading. Similarly, ICDAR 2015 [22] and CUTE-80 [38] also include rotated, curved, and blurred text, which has additional difficulties.

Lastly, COCO-Text [39] stands out for being the second-largest dataset, in terms of number of samples, with 63,686 images, as well as a high level of labelling. The annotations in this dataset include data such as whether the text is readable or not, the language of the text, that is, English or Non-English, and if it is artificially added text. To verify the robustness of the methods, images that do not contain any text are also included in this dataset, which is rare in similar datasets. Despite this, it remains as one of the least widely-used datasets, most notably in the recognition task [30].

5 | TEXT SPOTTING

5.1 | Methodologies

The Text Spotting task consists of the joint application of text detection and recognition. Depending on the type of system, they can be carried out jointly or separately [49], sharing information between them to improve their results [14]. In the implementation of both tasks, the use of deep learning architectures predominates. The most common methodologies for text spotting are step-by-step and integrated methodology [12, 30]. Both are depicted in Figure 5 [33].

The step-by-step methodology is divided into four phases: location, verification, segmentation and recognition. It consists of proposing candidate regions of text, analysing their uniform patterns or properties, separating the groupings of characters and finally transcribing them. The main advantage of this methodology is that the background of the image is filtered and the inclination in the localisation phase is estimated, which reduces computational cost in addition to processing oriented text [1]. This methodology is useful when processing both multi-language and multi-oriented text. Its main disadvantage is the complexity of integrating the various techniques required in each phase. Furthermore, adjustment of the parameters can affect multiple phases at the same time, making it difficult to locate the text correctly [1].

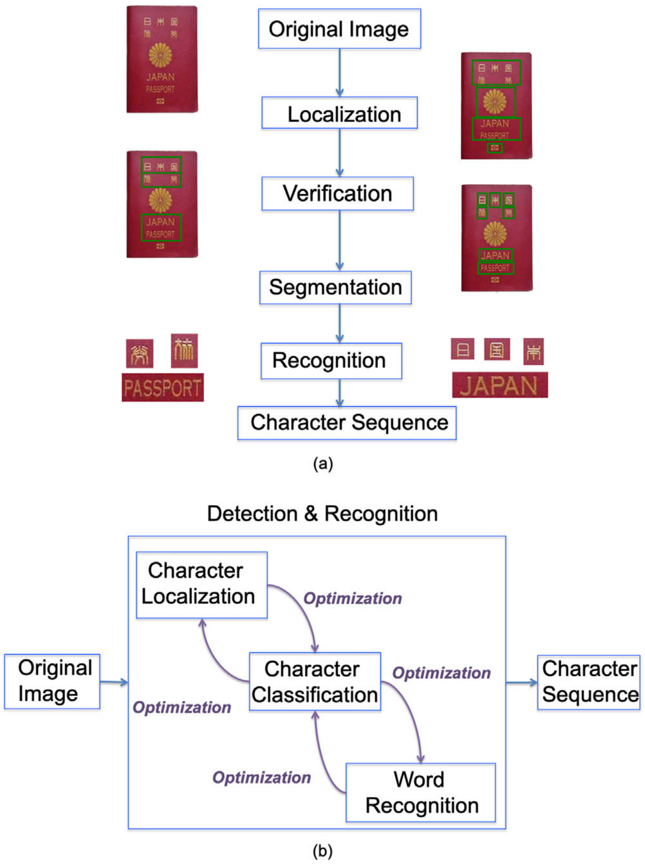

FIGURE 5 Step-by-step (a) and Integrated (b) methodologies applied to an image crawled from a website focused on document falsification

The integrated methodology combines the detection and recognition phases, sharing information between both tasks. Its main objective is to identify specific words using language and character models, avoiding the segmentation step of the integrated methodology. This makes the algorithms that implement this methodology less sensitive to low resolution text or complex backgrounds [1].

5.2 | Detection

In the step-by-step methodology, detection involves locating the regions of an image where there might be text, known as candidate regions. These regions are then verified by analysing their characteristics and classifying them according to whether they contain text or not. The main text detection methods can be classified as traditional or deep learning-based [2].

5.2.1 | Traditional methods

Traditional text detection methods extract low-level features, designed manually to represent text properties. They can be divided into two types: those based on regions and those for connected component analysis [1, 2].

In the region-based methods, it is determined which of the candidate regions obtained contains text using graph methods [58] and multiple-scale morphological operations [1]. This process is characterised by its simple architecture, which has the disadvantage of having a high computational cost when it is necessary to classify a large number of windows [1].

Methods based on connected components consist of graph algorithms where groups of image components are labelled using features such as colour similarity and their spatial distribution. Techniques such as Stroke Width Transform (SWT) [59] and Maximally Stable Extremal Regions (MSERs) [60] are used to search for candidate character regions, which eventually combine to form full body text. All the connected components methods are known as ’bottom-up’. Their main disadvantage is the high number of steps, which increases the computational cost [1]. Furthermore, due to the complexity of separating characters from languages not based on Latin alphabets, they do not perform well in multi-language texts [61].

To overcome the limitations of traditional methods [12], state-of-the-art works use deep learning [12, 23].

5.2.2 | Text spotting with deep learning

Due to the development of Deep Learning technologies, text detection methods have increased their performance by using CNNs [12], which allow text patterns to be extracted from images with results superior to traditional methods [30].

Despite advances in traditional methods, there are several outstanding problems that pose a challenge to state-of-the-art methods [5].

Yang et al. [62] highlighted the problems of variable scale and multi-orientation, designing a detection system based on the *Inception* [63] architecture and the deformable Position Sensitive RoI pooling (PSRoI) technique to detect curved text. In the case of multi-language text, special characters must be taken into account before deciding which object detection methods to use [64].

Ye et al. [65] tackle the issue of oriented text by fusing multiple-level features. They attempt to solve the issues of previous detectors by extracting global-level features and fusing them with character and word features into a single weakly supervised architecture.

Zhang et al. [66] also focused on the problem of multi-oriented text. They combined a deep CNN with a Graph Convolutional Network to form an end-to-end trainable network. Their approach divides text instances into components, estimating the geometry attributes using deep relational reasoning with the graph network before grouping the final text regions, which obtains good results on multi-oriented and multi-language text.

Complexity reduction in text detection is one of the most salient problems that methods try to solve, as well as reduced computational cost. Zhou et al. [67] proposed a two-phase-only detection procedure that uses a fully convolutional network (FCN) [68] to detect words or lines of text, excluding intermediate steps using multiple channels pixel-level features. The resulting areas are filtered using non-maximum suppression before obtaining the final regions. Liao et al. [69] presented a trainable text detection network called TextBoxes, which allows detecting text in scenes with high precision and efficiency without requiring any post-processing of the regions. Similarly, Tian et al. [70] introduced a weakly supervised text detection method with the aim of reducing the high amount of time spent preparing character labelling. The proposed method constitutes a robust detector from a set of smaller annotated images, which reduces computational time.

Text segmentation at the word or character level is a problem that can make subsequent text recognition more difficult. Deng et al. [71] improved the performance of this task by uniting pixels from the same instance and then detecting text in that instance. Its implementation is based on the use of a neural network to make predictions of pixels and text-containing regions, joining those classified as correct.

For low-resolution images, Tiang et al. [72] proposed a connection network that locates text sequences in convolutional layers using sliding window approaches. This method overcomes the limitations of bottom-up methods and obtains state-of-the-art results in ICDAR 2013 and 2015 [22, 35] datasets, using a vertical anchoring mechanism to predict the location of the regions. Compared to other methods, it stands out for its better performance and processing speed, being more imprecise in the case of vertical text. Despite this, the method fails in complex cases, such as image oversaturation or wide spaces between characters.

Lastly, Liao et al. [51] focused on segmentation-based approaches due to their performance in curved and oriented text. They proposed a module named Differentiable Binarization, which performed binarization in a segmentation network, making the whole process trainable in an end-to-end CNN with a light-weight architecture. Their proposal improved text detection without sacrificing computational cost.

5.2.3 | Detection method comparison

Table 3 shows a comparison of 20 detection methods based on their results on the most commonly used datasets, retrieved from their original publications. TextFuseNet [65] obtained the best results in all three datasets in Precision and F-1 Score, due to its focus on three-level features to improve oriented text detection.

TABLE 3 Summary of the results obtained by text detection methods – including their acronyms – in ICDAR 2013, ICDAR 2015 and SVT datasets. P, R and F-1 indicate precision, recall and f-score, respectively. '-' indicates results not reported

	ICDAR 2013			**ICDAR 2015**			**SVT**		
Publication	**P**	**R**	**F-1**	**P**	**R**	**F-1**	**P**	**R**	**F-1**
SDTL [26]	94.80	76.40	84.20	-	-	-	65.10	59.90	62.40
CTPN [72]	93.00	83.00	88.00	74.00	52.00	61.00	-	-	-
WeText [70]	91.10	83.10	86.90	-	-	-	75.70	49.70	59.80
SSTDRA [73]	89.00	86.00	88.00	80.00	73.00	77.00	-	-	-
WordSup [74]	93.34	87.53	90.34	79.33	77.03	78.16	-	-	-
R2CNN [75]	93.55	82.59	87.73	85.20	79.68	82.54	-	-	-
EAST [67]	-	-	-	83.27	78.33	80.72	-	-	-
TextBoxes [69]	89.00	83.00	86.00	-	-	-	-	-	-
TS-CRNN [76]	-	-	-	91.40	80.50	85.60	-	-	-
CSDN [77]	90.00	83.00	86.00	79.00	65.00	71.00	-	-	-
MFCN-CIAS [78]	93.00	79.00	85.00	76.00	54.00	63.00	-	80.00	-
MPSTP [79]	-	-	-	-	**96.16**	-	-	**88.41**	-
FOTS [49]	-	-	92.82	91.85	87.92	89.84	-	-	-
InceptText [62]	-	-	-	93.80	87.30	90.50	-	-	-
PixelLink [71]	88.60	87.50	88.10	85.50	82.00	83.70	-	-	-
FTSN [80]	-	-	-	88.60	80.00	84.10	-	-	-
DRRGN [66]	-	-	-	88.53	84.69	-	-	-	-
DB [51]	-	-	-	91.80	83.20	87.30	87.10	82.50	84.70
TextFuseNet [65]	**96.50**	**92.30**	**94.30**	**94.70**	89.70	**92.10**	**89.00**	85.30	**87.10**
PAN++ [81]	-	-	-	91.40	83.90	87.50	-	-	-

In the recall measure, MPSTP [79] achieved the best result with more than 8% over the second method on ICDAR 2015 and 3% in the SVT dataset.

5.3 | Recognition

The recognition task is more complex in real scenes than in digital documents [1, 12] due to issues such as the background on which the text is located. In documents, it is usually homogeneous [58, 82], allowing the use of binarization methods to segment characters, which does not usually happen in real scenes. Under special circumstances, such as blurry images of documents, these types of text can also present issues similar to those of real-scene images [83]. In real-scene images, the task becomes more complex due to problems such as noise, different backgrounds, variable text size, partial occlusion of various image regions, and the multiple orientations in which the text appears [2].

To solve these problems, Fully Convolutional Networks (FCN) [68] as well as various object detection architectures [84], such as Faster-RCNN [85] and YOLO [86], allow complete image processing, producing a pixel-level tagging that helps to identify objects or text regions of interest. Lastly, for multi-oriented or curved text recognition, it is common to use rectification networks, which allow correcting the orientation of the text found in an image [13, 20, 27].

The text recognition task is divided into two phases: the segmentation of the detected regions into characters and their transcription into readable strings [1]. Depending on the length of the final string to be transcribed, the methods can be organised at character, word or sequence level [25].

Segmentation is one of the most complex steps in the creation of recognition systems, allowing the components of the words to be separated individually for subsequent transcription. The methods proposed in recent years choose to integrate this step into the transcript [12].

Within the last step of recognition, the final character string prediction, is a new method split [12]. The first method is the use of CTCs (Connectionist Temporal Classification) which predicts sequences of arbitrary length. Subsequently, characters are obtained by removing empty regions and repeated characters [87]. The second type consists of systems based on attention mechanisms, which capture the information of the initial character sequence to predict the final transcription, by learning characteristics at the character level [88]. The segmentation produced by these techniques generally does not allow individual lines of text to be separated, so additional processing is required after execution.

Improving attention-based approaches, Litman et al. [89] proposed a method based on a stacked block architecture to refine predictions through repeated processing, enhanced with intermediate supervision and BiLSTM decoders.

Shi et al. [90] used a combination of DCNN + RNN network architectures simultaneously so that they process multidimensional images, predicting characters or words of different lengths, avoiding loss of information resulting from image rescaling, and generalising the method to the multi-language problem.

Xie et al. [91] proposed an alternative to these methods called Aggregation Cross-Entropy, a new loss function that allows methods to be adapted to two-dimensional recognition and to reduce their computational cost.

Shi et al. [14] proposed an architecture that uses a neural network model made up of a rectification network and a recognition network. The former corrects the orientation and other irregularities of the text, while the latter makes a prediction of character sequence from the corrected image. This structure allows processing a wide variety of textual irregularities, used in conjunction with detection systems.

To enhance the transcription step, Yue et al. [92] analysed the decoding process of attention-based decoders. They found that character-level decoders took not only contextual information but also positional features. Their proposal uses a dynamic ratio between positional and contextual cues to decode sequences in a robust way, achieving state-of-the-art results in both regular- and irregular-styled text datasets without significant performance drops.

One of the most common problems in text recognition is the various conditions that text can appear in, particularly rotation or non-straight text. Yan et al. [93] proposed a method based on primitive representations created by feature maps used as nodes of undirected graphs for visual representation, solving the problem of misaligned text.

To overcome the issues with irregular text, the most recent methods attempt to enhance their performance by including both visual and semantic features. Zheng et al. [94] propose a method that captures both visual and semantic distances between characters to improve recognition. [95] proposed a multi-stage and multi-scale decoder that jointly carries out visual and semantic reasoning, with recurrent stages refining the results. Lastly, [96] proposes a graph-based solution that groups pixels in a character instance based on their location similarity, achieving state-of-the-art results.

5.3.1 | Recognition method comparison

The performance of the recognition methods can be seen in Table 4. The most prominent method in ICDAR 2013 dataset is S-GTR [96] with its graph-based convolutional network and visual semantic knowledge, obtaining an accuracy of 97.80%. On ICDAR 2015, this method was also the most successful approach, obtaining a score of 87.30%. Furthermore, it generalises well on multilingual based datasets.

ESIR [20] and ASTER [14] obtained the same precision in the IIIT5K dataset using dictionaries, with scores of 99.60% in the case of the 50 words dictionary, and 98.80% on the 1000 lexicon. However, when no dictionary was used, S-GTR outperformed both on this dataset with an accuracy of 97.50%. Lastly, ASTER obtained the highest precision of 99.20% in the SVT dataset, but S-GTR also outperformed it in the lexicon-free category with a precision of 95.80%. As seen in the table, the recognizers that focus on irregular text and oriented stand out from the rest in both regular and irregular datasets.

5.4 | End-to-end methods

The methods that integrate both tasks (i.e., detection and recognition) in the same system are known as end-to-end. In this type of system, both detection and recognition can share information among themselves [14, 69].

The main classification of this type of system is based on whether they implement separate models in each task, which can accumulate errors from each phase, or if they integrate both into a single trainable network [12]. The latter type of system benefits from the feedback between both tasks. When run separately, no common characteristics are shared for both tasks that could improve their performance [49].

Gupta et al. [26] trained a fully convolutional regression network for text detection, combining it with a word classifier for recognition. The detection network makes a prediction at all locations in the image, to classify them into text containment parameters. Despite this feedback between tasks, the collected methods highlight the greater relevance of detection in recognition [5]. Some approaches allow for the generation of candidate regions that are then refined using a separate recognition model [14, 69]. In the same way, the results of recognition make it possible to reduce the number of false positives in detection by using strongly contextualised dictionaries [13].

Running both tasks simultaneously can lead to problems such as the accumulation of errors due to detection or erroneous transcripts, increasing the complexity of systems by acquiring a greater number of parameters, or the difficulty of training both tasks together. The most prominent methods try to solve these problems in various ways. Li et al. [76] proposed a system that combines multiple convolutional layers, a network of candidate text regions, and a recurring neural network. The method avoids the need to group characters or separate lines of text, avoiding error accumulation.

Liu et al. [49] featured a simultaneous run of detection and recognition, creating a trainable system using convolutional neural networks shared between detection and recognition. The goal of this system is to reduce the number of complex post-processing steps and parameter adjustment. Bartz et al. [97] presented a method that used a single deep neural network that jointly trained detection and text recognition in a semi-supervised manner. However, this increased the difficulty of training the system and did not obtain good results on incident text.

For arbitrarily shaped text, Wang et al. [81] proposed an efficient method based on pixel representation. Their approach detects text as a kernel, which is surrounded by peripheral pixels, distinguishing text lines from the background using

TABLE 4 Summary of the precision results obtained by text recognition methods—including their acronyms—in the ICDAR 2013, ICDAR 2015, SVT, and IIIT5K datasets. None, 50 and 1k indicate the size of lexicons used. * indicates the F-1 score reported, instead of the precision

	ICDAR 2013	ICDAR 2015	IIIT5K			SVT	
Publication	**None**	**None**	**50**	**1k**	**None**	**50**	**None**
CRNN [90]	86.70	-	97.60	94.40	78.20	96.40	80.80
RARE [27]	88.60	-	96.50	93.80	81.90	96.10	81.90
STN-OCR [97]	90.30	-	-	-	86.00	-	79.80
TS-CRNN [76]	-	92.42*	-	-	-	84.91*	66.80*
ASTER [14]	91.80	76.10	**99.60**	**98.80**	93.40	**99.20**	**93.60**
AON [98]	-	68.20	**99.60**	98.10	87.00	96.00	82.80
FOTS [49]	93.90*	82.39*	-	-	-	-	-
SAR [99]	94.00	78.80	99.40	98.20	95.00	98.50	91.20
MORAN [13]	92.40	68.80	97.90	96.20	91.20	96.60	88.30
ESIR [20]	91.30	76.90	**99.60**	**98.80**	93.30	97.40	90.20
ACE [91]	89.70	68.90	-	-	82.30	-	82.60
SCATTER [89]	94.70	82.80	-	-	93.90	-	92.70
Baek [15]	92.30	71.80	-	-	87.90	-	87.50
RobustScanner [92]	94.80	79.20	-	-	95.40	-	89.30
Pren2D [93]	96.40	83.00	-	-	95.60	-	94.00
JVSR [95]	95.50	84.00	-	-	95.20	-	92.20
ABINET [100]	97.40	86.00	-	-	96.20	-	93.50
S-GTR [96]	**97.80**	**87.30**	-	-	**97.50**	-	**95.80**
CDistNet [94]	97.67	86.25	-	-	96.57	-	93.82

Pixel Aggregation. For recognition, they use an attention-based recognition head, achieving a computationally efficient end-to-end Text Spotter. Their approach also highlights the different possible configurations for the different stages, showcasing higher scores with VGG as the backbone, but slowing performance.

The combination of real-time end-to-end word recognition and curved Text Spotting is the main focus of the work proposed by Liu et al. [101]. Their approach is based on a parameterized Bézier curve, which allows for efficient recognition of curved text. This Bézier-based detection is then combined with a simplified version of Convolutional Recurrent Neural Networks (CRNN), which keeps both tasks separate so that the detection process does not directly affect recognition.

Multi-language text is a prominent problem for end-to-end methods. Wu et al. [61] proposed a method focused on the Chinese alphabet, which presents additional difficulties in not being categorised at the word level due to the language context, as well as the length and orientation of its characters. Its proposal consists of the combination of three modules: Chinese character detector, keyword locator, and keyword extractor. The results obtained were analysed by searching for keywords, comparing their scale and distance to documented words.

Comparison between different recognition approaches can be complex due to their specific goals, training datasets and multiple configurations, as well as the lack of a uniform protocol for recognition evaluation [30]. Baek et al. [15] highlighted the issues with comparison inconsistencies due to using different datasets for testing and training across state-of-the-art approaches. Using the VGG [102] and ResNet [29] architectures, they analysed the use of transformation techniques, architectures and decoders in terms of accuracy, speed and memory, resulting in a framework that can be applied to most methods to measure their performance on two synthetic datasets.

6 | IMPLEMENTATIONS

Convolutional Neural Networks constitute the most common architectures in text recognition and detection [23]. The most used networks by the methods collected in this study are VGG-Net [102] and ResNet, [29], more specifically VGG16 and ResNet-50. Both architectures can be used together, as seen in [62], where the authors combine two ResNet101 networks, two ResNet-50 networks, and a VGG network with the objective of performing multi-orientation text detection.

Other approaches have focused on the application of alternative CNNs. Zhou et al. [67] tried to find an alternative to using VGG, proposing the use of PVANET [103] and achieving state-of-the-art results on the MSRA-TD500 dataset. Yan et al. [93] used EfficientNetv3 [104] as their feature extraction model. Busta et al. [105] chose to adapt the YOLOv2 architecture [106] due to its higher precision and lower complexity compared to

TABLE 5 Backbone-based architectures reported in the surveyed methods

Architecture	Detectors	Recognizers	End-to-end
ResNet [29]	4	10	4
VGG [102]	11	4	4
Others	1	1	1

other networks. This choice is justified by the need in VGG for more than 30 trillion operations to process images of size 224 × 224.

Similarly, multiple approaches have proposed several configurations using variations of these standalone architectures [51, 65], such as ResNet-18 or ResNet-101, analysing their impact on floating point operations and the balance between speed and performance. Table 5 summarises the most common architectures used on the surveyed methods for both detectors and recognizers, highlighting ResNet in the case of recognizers over the use of VGG, which is more common in detectors.

6.1 | VGG-Net

The VGG-Net architecture [102] is composed of 16 convolutional layers, increasing the depth of CNN at the time by adding a greater number of layers. Most of the reviewed methods modify the base structure of VGG to adapt it to particular problems, including reducing the computational cost or the generated models [73, 75]. To do this, the models are trained in various datasets such as SynthText, ICDAR [74] or ImageNet [72, 107].

Deng et al. [71] used VGG16 as their main structure to make comparisons with similar publications. Their study highlights that other deep learning models and architectures should be investigated for better computational performance and cost. Gupta et al. [26] carried out a similar study using VGG-Net pre-trained models, finding that the smaller models produce very similar results compared to the larger models, reducing the computational time required.

Modifications vary according to the particular objective of each method. Wu et al. [61] used VGG-16 to acquire high-quality image characteristics, detecting text at multiple scales. Tian et al. [70] applied semi and weakly supervised learning in a pre-trained VGG-16 network, generating a smaller model by fine-tuning the network parameters. Liao et al. [69] extended VGG-Net by designing a network of 28 fully convolutional layers, 13 inherited from VGG-16, after which nine are added for better results. Li et al. [76] used a CNN network derived from the VGG-16 network, eliminating the five layers of max-pooling connected components.

6.2 | ResNet

The ResNet (Residual Network) architecture [29] arises from the need to train deep neural networks in a simple and efficient way. This architecture contains a total of 152 layers, that is, eight times greater than VGG-Net, but with less complexity due to its design aimed at ease of optimisation without reducing the depth of the network.

Liu et al. [49] used ResNet-50 as their backbone for text detection, obtaining state-of-the-art results. Ye et al. [65] combined the use of ResNet-18 and ResNet-50 on different datasets for their proposal, analysing which architecture obtained the best performance and computational cost.

Bartz et al. [97] used the ResNet-50 architecture after discovering that their proposed system had a faster learning period and higher performance compared to other structures. Its conclusion justifies the use of the residual blocks that make up ResNet, which retain a greater amount of information in the first convolutional layers, thus improving the results obtained.

6.3 | Libraries and coding language

Only half of the 40 methods collected in this study provide details on the specifications of the software used. The reviewed studies specify the hardware equipment employed to facilitate the reproducibility of the results, highlighting the use of Ubuntu 14.04 distributions. Graphics card information and CPU or multiple GPU usage are detailed. Among the methods analysed, 13 used TITAN cards and 13 used TESLA.

Regarding the programming language, the use of the *C*++ programming language over *C*, Python, or MATLAB as the main implementations stands out, justified by the objective of building real-time systems. Of the methods surveyed, 17 did not indicate details on their software implementation. The most common libraries in the collected methods are OpenCV, TensorFlow [108] and Caffe [109]. In Python environments, PyTorch [110] is one of the most widely used machine learning libraries, being present in 9 of the methods surveyed.

Table 6 presents the most relevant information collected from the methods. We excluded parameter-based information such as training epochs or loss functions, as it can further complicate comparing the real performance of the methods.

The methods that do detail software specifications [13, 71] do not indicate the version of the languages or libraries used or their justification, except for those that emphasise the importance of computational efficiency. Because of this, comparison between methods is still limited to various training and specific conditions.

7 | CHALLENGES OF TEXT SPOTTING METHODS

Scene text detection methods present various issues that make it difficult to improve both the performance of the method and its implementation in real-time-based systems. Our review presents the current tendencies and problems of text detection and recognition. Following our analysis, we classified these

TABLE 6 Comprehensive data of the text spotting methods analysed

Method	Year	Code	Problem	D	R	End-to-End	Performance	Software	Hardware	Methods	Architectures	Datasets
RARE [27]	2016	No	Irregular-curved text.	No	Yes	No	2ms per image	-	Intel Xeon(R) E5-2620 2.40GHz CPU, NVIDIA GTX-Titan GPU, and 64GB RAM	TPS, BiLSTM, GRU cells	-	IC-13, SVT-P, CUTE-80, IIIT5K
SDTL [26]	2016	Yes	Multiscale text detection.	Yes	Yes	Yes	15 images per second in GPU.	-	-	CNN	VGG-16	IC-11, IC-13, SVT, SynthText
CTPN [72]	2016	Yes	Localization of text sequences in convolutional layers.	Yes	No	No	0,14s per image	Caffe	-	RPN, RNN, BLSTM	VGG-16	IC-11, IC-13, IC-15, Multilingual
FTSN [80]	2017	No	Multi-oriented text detection. Generalization and flexibility.	Yes	No	No	-	MXNet	Intel i7 6700K CPU, 64GB RAM, GTX 1080	Deep CNN Model RPN PSROI Pooling NMS	ResNet-101	IC-15, MSRA-TD500, TotalText, SynthText
WeText [70]	2017	No	Semi supervised learning with less data required.	Yes	No	No	0,32s per image	-	Titan-Xp GPU	SSD Framework, NMS	VGG-16	IC-13, COCO-Text, SynthText
SSTDRA [73]	2017	No	Suppress background interference. Multiple scales and orientation.	Yes	No	No	0,13s per image in GPU	Caffe	Titan-Xp GPUs	SSD Framework, NMS	VGG-16	IC-13, IC-15, COCO-Text
WordSup [74]	2017	No	Character-level detector. Multiple languages.	Yes	No	No	500ms for a 500x500 image in GPU	-	Nvidia Tesla K40 GPU	Character detector and text structure analysis pipeline. FCN, FPN, NMS,	VGG-16	IC-13, IC-15, COCO-Text, Synthtext
R2CNN [75]	2017	No	Arbitrarily oriented text in natural scenes.	Yes	No	No	0,4s per image	-	Tesla K80 GPU	CNN and Feature maps, RPN, Roi Pooling, Inclined NMS	VGG-16	IC-13, IC-15
EAST [67]	2017	Yes	Step reduction, improving results in complex scenes.	Yes	No	No	13,2 fps for 720p resolution in GPU	-	NVIDIA Titan X graphic and Intel E5-2670 v3 @ 2.30GHz	FCN, RBOX, QUAD, NMS	VGG-16, PVANET	IC-15, MSRA-TD500, COCO-Text
TextBoxes [69]	2017	Yes	Word region prediction. Reduce intermediate steps.	Yes	No	No	0,09s per image	-	Titan X GPU	FCN, NMS,	VGG-16	IC-11, IC-13, SVT, SynthText
Deep TextSpotter [105]	2017	Yes	Simultaneous detection and recognition. Adapts to the shape and resolution of the text.	Yes	Yes	Yes	10 fps on GPU	-	NVIDIA K80 GPU	FPN, RPN, Bilinear Sampling, CTC	YOLO-V2	IC-13, IC-15, COCO-Text
STN-OCR [97]	2017	Yes	Semi-supervised deep neural network.	Yes	Yes	Yes	-	MXNet	Intel(R) Core(TM) i7-6900K CPU. 64 GB RAM and 4 TITAN X (Pascal) GPUs	DNN, Feed Forward CNN with a RNN, BLSTM	ResNet	Others
TS-CRNN [76]	2017	No	Avoid intermediate steps in Text Spotting.	Yes	Yes	Yes	0,90s per image in GPU	-	NVIDIA Tesla M40 GPU con 24GB	CNN, RoI pooling, LSTM	VGG-16	IC-11, IC-15, SVT, SynthText
CSDN [77]	2017	No	Predicts oriented regions on the word level.	Yes	No	No	450ms for a 1000x560 image in GPU	Caffe	3.3Ghz 6-score CPU, 32G RAM, GTX Titan X GPU	FCN	VGG-16	IC-13, IC-15, SynthText
MFCN-CIAS [78]	2017	No	Curved and oriented text.	Yes	No	No	-	-	NVIDIA	Multi-Scale FCN, CNN	VGG-16	IC-13, IC-15, SVT, CUTE-80

(Continues)

TABLE 6 (Continued)

Method	Year	Code	Problem	D	R	End-to-End	Performance	Software	Hardware	Methods	Architectures	Datasets
MPSTP [79]	2017	No	Max pooling based region grouping for text extraction.	Yes	No	No	-	MATLAB, C++	CPU Intel Xeon(R) CPU ES-1650v2-3.5 GHz x 12 and 32 GB Ram	CNN	-	IC-15, MSRA-TD500, SVT
CRNN [90]	2017	Yes	Recognition of text with arbitrary length without segmentation.	No	Yes	No	0,16s per image on IC-03 without lexicon	Torch7/Cuda , C++	2.50 GHz Intel(R) Xeon(R) E5-2609 CPU, 64 GB RAM and NVIDIA(R) Tesla(TM) K40 GPU	CRNN, LSTM, CTC	VGG	IC-13, SVT, IIIT5K, SynthText
KSCMO [61]	2018	Yes	Multi-oriented detection of Chinese characters.	Yes	Yes	Yes	-	-	-	SSD	VGG-16	Others
AON [98]	2018	Yes	Irregular and curved text.	No	Yes	No	630 samples per second in testing	CUDA 8.0 and CUDNN v7	Intel Xeon(R) E5-2650 2.30GHz CPU, NVIDIA Tesla P40 GPU and 128GB RAM	Filter Gate. Bidirectional LSTM	-	IC-15, SVT, SVT-P, IIIT5K, CUTE-80
ASTER [14]	2018	Yes	Irregular, distorted and curved text.	No	Yes	No	-	TensorFlow	NVIDIA TITAN Xp with 12GB	CTC, BLSTM	VGG, ResNet	IC-13, IC-15, SVT, SVT-P, SynthText, IIIT5K
FOTS [49]	2018	No	Incidental and oriented text.	Yes	Yes	Yes	22,6 fps	Caffe (Modified)	SSS Titan-Xp GPU	FCN, ROI-Rotate, NMS	ResNet-50	IC-13, IC-15, IC-17 MLT, SynthText
IncepText [62]	2018	No	Incidental, multi-oriented and multi-scale text.	Yes	No	No	Computational cost of 20ms	-	Nvidia Tesla M40 GPU	Instance-Aware Segmentation, Deformable PSROI pooling	ResNet-50	IC-15, MSRA-TD500, RCTW-17
PixelLink [71]	2018	Yes	Word separation.	Yes	No	No	7,3 fps in IC-15	OpenCV, Tensorflow, Python	GTX Titan X	Instance Segmentation	VGG-16	IC-13, IC-15, MSRA-TD500, SynthText
SAR [99]	2019	Yes	Irregular text with a smaller sized model.	No	Yes	No	-	Torch	NVIDIA Titan X Gpu with 12GB Memory	LSTM + Attention Module	ResNet	IC-13, IC-15, SVT, SVT-P IIIT5K, CUTE-80, SynthText, COCO-Text
MORAN [13]	2019	Yes	Irregular text. Weakly supervised approach.	No	Yes	No	10,4ms on a five-character image without lexicon	PyTorch, CUDA 8.0, CuDNN v7	NVIDIA GTX-1080Ti GPU	CNN-BLSTM	-	IC-13, IC-15, SVT, SVT-P, IIIT5K, CUTE-80, SynthText
ESIR [20]	2019	No	Curved and distorted text. Robust rectification.	No	Yes	No	28ms per image	TensorFlow	Intel Core i7- 7700K CPU, NVIDIA GeForce GTX 1080 Ti with 12GB memory and 32GB RAM	BiLSTM	VGG, ResNet	IC-13, IC-15, SVT, SVT-P, IIIT5K, CUTE-80, SynthText
ACE [91]	2019	Yes	Optimized loss function, easy to implement.	No	Yes	No	30 times faster than CTC	-	NVIDIA TITAN X with 12GB Memory	CNN-LSTM, FCN	ResNet-101	IC-13, IC-15, SVT, SVT-P, IIIT5K, CUTE-80, SynthText
ABCNET [101]	2020	Yes	Real-time Text Spotting on curved text	Yes	Yes	Yes	22.8 fps on TotalText	-	4 Tesla V100 GPUs	RoiAlign, CTC loss	Resnet-50-FPN	COCO-Text, Total-Text

(Continues)

TABLE 6 (Continued)

Method	Year	Code	Problem	D	R	End-to-End	Performance	Software	Hardware	Methods	Architectures	Datasets
DRRGN [66]	2020	Yes	Arbitrary shape text detection	Yes	No	No	-	PyTorch 1.2.0.	Single RTX-2080Ti GPU	RoiAlign, NMS.	VGG-16 and FPN	IC-15, MSRA-TD500, Total-Text, IC-17, SynthText
SCATTER [89]	2020	No	Irregular text recognition	No	Yes	No	-	PyTorch	Tesla V100 GPU with 16GB memory.	TPS, CTC, BiLSTM	ResNet-29	IC-13, IC-15, SVT, SVT-P, IIIT5K, CUTE-80, SynthText, MJSynth
Baek et al. [15]	2019	Yes	Model analysis and stage division configuration	No	Yes	No	27.6 ms on the best-performance configuration	PyTorch 1.3.1,	Intel Xeon(R) E5-2630 v4 2.20GHz CPU, NVIDIA TESLA P40 GPU. 252GB of RAM.	TPS, CTC, BiLSTM	VGG, ResNet and RCNN	IC-13, IC-15, SVT, SVT-P, IIIT5K, CUTE-80, SynthText, MJSynth
RobustScanner [92]	2020	No	Attention-based encoder-decoder enhancements on contextless texts	No	Yes	No	-	PyTorch	4 NVIDIA Titan X GPUs with 12 GB memory.	LSTM+Attention Module	ResNet	IC-13, IC-15, SVT, SVT-P, IIIT5K, CUTE-80, SynthText, MJSynth
DB [51]	2020	Yes	Binarization enhancements for segmentation-based detection methods	Yes	No	No	62 FPS on MSRA-TD500	PyTorch 1.2	NVIDIA 1080ti GPU	Deformable Convolution, Differentiable Binarization	ResNet-18, ResNet-50	IC-15, MSRA-TD500, Total-Text, IC-17, SynthText
TextFuseNet [65]	2020	Yes	Feature-fusion for arbitrary text detection	Yes	No	No	4.1 FPS on ICDAR2015	-	NVidia Tesla V100 (16G) GPUs	FPN, RPN	ResNet-50, ResNet-101	IC-13, Total-Text, SynthText, TotalText
PAN++ [81]	2021	Yes	Efficient text spotting of arbitrary shapes in natural scenes.	Yes	Yes	Yes	29.2 FPS on end-to-end tasks on TotalText	PyTorch	NVIDIA 1080Ti GPU	Masked RoI, Recognition Head, FPEM (improved), PA (improved)	VGG16, ResNet18, ResNet50	IC-13, MSRA-TD500, Total-Text, IC-17, SynthText, COCO-Text, RCTW-17
Pren2D [93]	2021	Yes	Multi-oriented scene texts	No	Yes	No	67.4ms average on single image	PyTorch	Tesla V100	Graph Convolutional Networks	EfficientNet-B3	MJSynth, SynthText, IIIT5K, SVT, IC13, IC15, SVT-P, CUTE-80.
JVSR [95]	2021	No	Visual and semantic information for text recognition	No	Yes	No	-	PyTorch	NVIDIA RTX-2080-Ti GPU	FPN, RNN	ResNet	SynthText, IIIT5K, SVT, IC13, IC15, SVT-P, CUTE-80,
ABINET [100]	2021	Yes	Application of linguistic knowledge in text recognition	No	Yes	No	-	-	NVIDIA 1080Ti GPUs, NVIDIA Tesla V100	Visual and Language Models. Multi-level attention	ResNet and Transform-ers	MJSynth, SynthText, IIIT5K, SVT, IC13, IC15, SVT-P, CUTE-80
S-GTR [96]	2021	No	Spatial context of visual semantics	No	Yes	No	18.8ms on average	-	NVIDIA V100 GPUs	FCN, Graph Convolutional Networks	-	MJSynth, SynthText, IIIT5K, SVT, IC13, IC15, SVT-P, CUTE-80
CDistNet [94]	2021	Yes	Visual and semantic features for irregular text	No	Yes	No	-	PyTorch	NVIDIA 3080 GPUs	TPS	ResNet45 and Transform-ers	MJSynth, SynthText, IIIT5K, SVT, IC13, IC15, SVT-P, CUTE-80

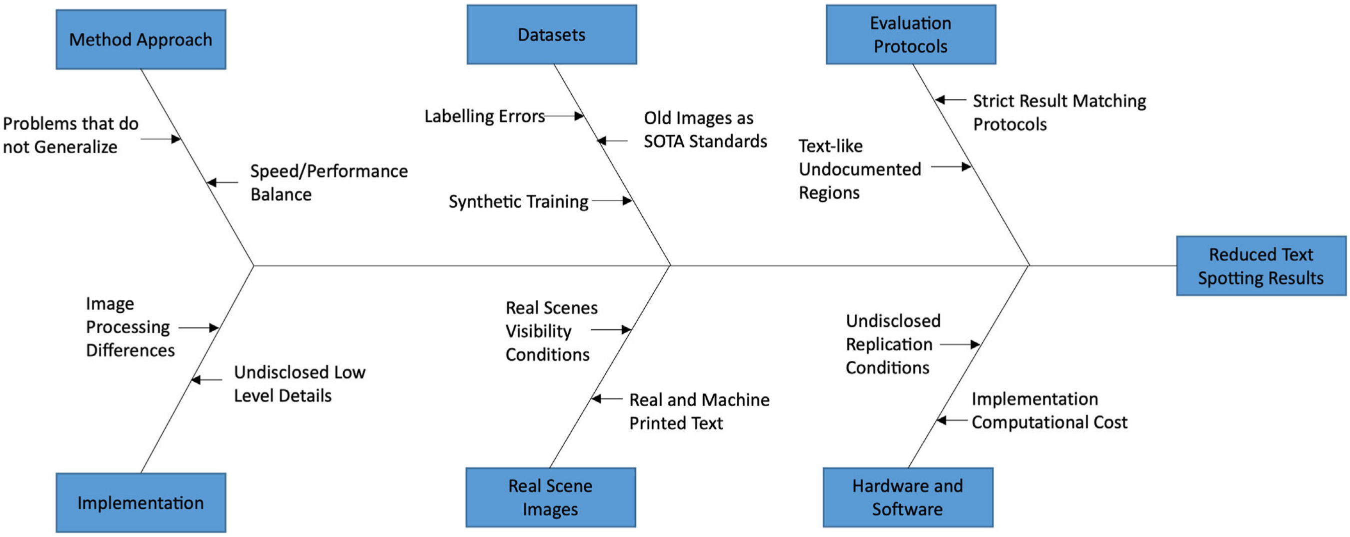
Method Approach
Datasets
Evaluation Protocols
Problems that do not Generalize
Speed/Performance Balance
Labelling Errors
Old Images as SOTA Standards
Synthetic Training
Strict Result Matching Protocols
Text-like Undocumented Regions
Reduced Text Spotting Results
Image Processing Differences
Undisclosed Low Level Details
Real Scenes Visibility Conditions
Real and Machine Printed Text
Undisclosed Replication Conditions
Implementation Computational Cost
Implementation
Real Scene Images
Hardware and Software

FIGURE 6 Fishbone summary of Text Spotting issues

problems into six categories; method approach, implementations, datasets, hardware and software, evaluation protocols and real-scene images. These issues are depicted in the fishbone diagram included in Figure 6.

For method approaches, we identified a strong focus on balancing speed and performance, with authors focusing on lightweight architectures. However, the specific problems that each method tackles make its generalisation difficult in most use cases. Due to this, methods need specific datasets, which are not as widespread as state-of-the-art datasets. Furthermore, method training in state-of-the-art approaches relies heavily on synthetically generated images, which may not generalise well on methods that tackle a specific problem due to not being able to filter by images relevant to the approach. In addition, methods that use custom-based datasets lack the sheer number of synthetically created images and frequently include labelling errors, which can penalise strict evaluation methods.

Despite the recent focus on real-scene based images, which may include several types of text, the most recent datasets are not as widespread as older images, which are favoured due to simpler evaluation protocols despite their outdated image conditions. Most of the extended datasets for evaluation do not include curved or multi-type text, a highlighted problem in recent approaches [14].

Evaluation protocols such as DetEval can also penalise the results of the current method. Due to strict IoU evaluations and string-matching criteria, methods can be heavily penalised on non-lexicon-based approaches for detecting text-like regions that are not documented as such or due to a mismatch on a similar character inside the final string. Less strict methods could be used in order to measure the performance of the method regarding false positives, as well as dataset upgrades to include text-like regions inside "don't care" categories.

Lastly, the hardware and software implementations for each approach are not often disclosed in most methods. These data are of relevance when trying to replicate the methods' speed and performance in order to compare approaches fairly in the same environments. Although hardware is more often disclosed, the computational cost of the chosen software implementations for each method would provide important information to enhance real-time system performance by comparing it with other available libraries.

8 | CONCLUSIONS

Text Spotting has been a topic of interest in recent years. Recent advances have focused on the multilingual and oriented text, as well as speed-performance balance. In this article, we compared 40 recent text detection and recognition methods, summarizing their approaches, techniques, and datasets used. These methods have also been analysed from a perspective not considered in similar studies, such as the inclusion of details related to hardware, software, and implementation libraries.

We also included architectures, the libraries employed, and their performance in three detection and four recognition datasets. From the reviewed methods, 15 of them focus on the detection task, ten prioritise recognition, and six are end-to-end systems that combine detection and recognition.

Our initial analysis reveals the lack of a unified evaluation protocol that consider the multiple method configurations and the different training datasets used. We pay special attention to methods focused on rectification due to the increasing presence of irregular text, such as S-GTR and ESIR. Additionally, we recognise Python and C++ as the main programming languages due to the libraries support and its focus on real-time applications, respectively, as well as the use of TITAN graphic cards amongst the collected methods.

We also discuss the most relevant methods, problems, architectures and implementation choices, which helps to establish a more fair baseline for method comparison. With their varying approaches, methods would benefit from a standardised style of report that takes into account their configurations.

We have also highlighted the main problems for each task. For detection, oriented text, incident text and the correct division of character lines. In the recognition task, the use of rectification networks allows correction of irregular text curvature and improvement of transcription quality. The unification of both tasks in end-to-end systems and the reduction of intermediate processes is also an important research line, looking for improving performance and computational cost in real-time systems, what makes them the recommended systems for applications in industrial environments.

Future challenges of this task include the support of different types of text (font, size, machine or real text). Although current methods are focused on incidental text, the most widely used datasets are outdated and their images are not representative enough of the most prominent current problems in the task. Additionally, newer evaluation protocols should be considered to reduce method penalization. Images in state-of-the-art datasets frequently include text regions that are not documented on the ground truth. When the methods detect these regions, they are labelled as false positives, decreasing their performance. Lastly, newer architectures and techniques (e.g., Transformers) that have surpassed CNNs on other computer vision related tasks are beginning to find applications in Text Spotting, enhancing the final text output.

In future work, we will include a higher number of methods oriented to recognition, as this is a task less represented by the collected methods, in favour of detection. Due to the difficulties encountered in comparing results, the methods would benefit from a standardised results presentation methodology, making the comparison more straightforward and accurate.

ACKNOWLEDGEMENTS

This work was supported by the grant “Ayudas para la realización de estudios de doctorado en el marco del programa propio de investigación de la Universidad de León Convocatoria 2018” and also by the framework agreements between the Universidad de León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 22 and Addendum 01. The authors acknowledge the support of NVIDIA Corporation with the donation of the Titan XP GPU used for this research.

CONFLICT OF INTEREST

The authors have declared no conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated during the current study.

ORCID

Pablo Blanco-Medina https://orcid.org/0000-0002-5768-113X
Eduardo Fidalgo https://orcid.org/0000-0003-1202-5232
Enrique Alegre https://orcid.org/0000-0003-2081-774X
Víctor González-Castro https://orcid.org/0000-0001-8742-3775

REFERENCES

1. Ye, Q., Doermann, D.: Text detection and recognition in imagery: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 37(7), 1480–1500 (2015)
2. Yin, X., Zuo, Z., Tian, S., Liu, C.: Text detection, tracking and recognition in video: a comprehensive survey. IEEE Trans. Image Process. 25(6), 2752–2773 (2016)
3. Greenhalgh, J., Mirmehdi, M.: Recognizing text-based traffic signs. IEEE Trans. Intell. Transp. Syst. 16(3), 1360–1369 (2015)
4. Ham, Y.K., Kang, M.S., Chung, H.K., Park, R.H., Park, G.T.: Recognition of raised characters for automatic classification of rubber tires. Opt. Eng. 34(1), 102–110 (1995)
5. Liu, X., Meng, G., Pan, C.: Scene text detection and recognition with advances in deep learning: A survey. Int. J. Doc. Anal. Recogn. (IJDAR) 22(2), 143–162 (2019)
6. Greenhalgh, J., Mirmehdi, M.: Detection and recognition of painted road surface markings. In: 4th International Conference on Pattern Recognition Applications and Methods, pp. 130–138. Springer, Cham (2015)
7. Schulz, R., Talbot, B., Lam, O., Dayoub, F., Corke, P., Upcroft, B., et al.: Robot navigation using human cues: A robot navigation system for symbolic goal-directed exploration. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1100–1105. IEEE, Piscataway (2015)
8. Blanco Medina, P., Fidalgo, E., Alegre, E., Vasco Carofilis, R.A., Jañez Martino, F., Villar, V.F.: Detecting vulnerabilities in critical infrastructures by classifying exposed industrial control systems using deep learning. Appl. Sci. 11(1), 367 (2021)
9. Yoon, Y., Ban, K.D., Yoon, H., Kim, J.: Automatic container code recognition from multiple views. ETRI J. 38(4), 767–775 (2016)
10. Blanco Medina, P., Fidalgo, E., Alegre, E., Jáñez Martino, F.: Improving text recognition in tor darknet with rectification and super-resolution techniques. In: IET Conference Publications. vol. 2019, pp. 32–37. Institution of Engineering and Technology, Stevenage (2019)
11. Fidalgo, E., Alegre, E., Fernández Robles, L., González Castro, V.: Classifying suspicious content in tor darknet through semantic attention keypoint filtering. Digital Invest. 30(5), 12–22 (2019)
12. Long, S., He, X., Yao, C.: Scene text detection and recognition: The deep learning era. Int. J. Comput. Vision 129(1), 161–184 (2021)
13. Luo, C., Jin, L., Sun, Z.: MORAN: A multi-object rectified attention network for scene text recognition. Pattern Recogn. 90, 109–118 (2019)
14. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: ASTER: An attentional scene text recognizer with flexible rectification. IEEE Trans. Pattern Anal. Mach. Intell. 41(9), 2035–2048 (2019)
15. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., et al.: What is wrong with scene text recognition model comparisons? Dataset and model analysis. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4714–4722. IEEE, Piscataway (2019)
16. Blanco-Medina, P., Fidalgo, E., Alegre, E., MAl-Nabki, W.: Detecting textual information in images from onion domains using text spotting. Actas de las XXXIX Jornadas de Automática 2018, 975–982 (2018)
17. Blanco Medina, P., Fidalgo, E., Alegre, E., Alaiz Rodríguez, R., Jáñez Martino, F., Bonnici, A.: Rectification and super-resolution enhancements for forensic text recognition. Sensors 20(20), 5850 (2020)
18. Al Nabki, M.W., Fidalgo, E., Alegre, E., Fernández Robles, L.: ToRank: Identifying the most influential suspicious domains in the tor network. Expert Syst. Appl. 123, 212–226 (2019)
19. Kazmi, W., Nabney, I., Vogiatzis, G., Rose, P., Codd, A.: An efficient industrial system for vehicle tyre (Tire) detection and text recognition using deep learning. IEEE Trans. Intell. Transp. Syst. 22(2), 1264–1275 (2021)
20. Zhan, F., Lu, S.: ESIR: End-To-End scene text recognition via iterative image rectification. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, pp. 2059–2068. IEEE, Piscataway (2019)
21. Wang, K., Belongie, S.: Word spotting in the wild. In: European Conference on Computer Vision, pp. 591–604. Springer, Berlin (2010)

22. Karatzas, D., Gomez Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., et al.: ICDAR 2015 competition on robust reading. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, pp. 1156–1160. IEEE, Piscataway (2015)
23. Lin, H., Yang, P., Zhang, F.: Review of scene text detection and recognition. Arch. Comput. Methods Eng. 1–22 (2019)
24. Husnain, M., Missen, M.M.S., Mumtaz, S., Coustaty, M., Luqman, M., Ogier, J.M.: Urdu handwritten text recognition: A survey. IET Image Proc. 14(11), 2291–2300 (2020)
25. Deng, D., Liu, H., Li, X., Cai, D.: Current trends in text spotting. In: 2017 International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1–6. (2017)
26. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2315–2324. IEEE, Piscataway (2016)
27. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4168–4176. IEEE, Piscataway (2016)
28. Jaderberg, M., Simonyan, K., Zisserman, A., kavukcuoglu, k.: Spatial transformer networks. In: Advances in Neural Information Processing Systems, vol. 28, pp. 2017–2025. Curran Associates, Inc., Red Hook (2015)
29. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE, Piscataway (2016)
30. Chen, X., Jin, L., Zhu, Y., Luo, C., Wang, T.: Text recognition in the wild: A survey. ACM Comput. Surv. 54(2), (2021)
31. Gupta, N., Jalal, A.S.: Traditional to transfer learning progression on scene text detection and recognition: A survey. Artificial Intelligence Review 55, 3457–3502 (2021)
32. Shahab, A., Shafait, F., Dengel, A.: ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on, pp. 1491–1496. IEEE, Piscataway (2011)
33. Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1083–1090. IEEE, Piscataway (2012)
34. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: BMVC - British Machine Vision Conference, pp. 127.1–127.11. BMVA, London (2012)
35. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., et al.: ICDAR 2013 robust reading competition. In: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on, pp. 1484–1493. IEEE, Piscataway (2013)
36. Phan, T., Shivakumara, P., Tian, S., Tan, C.: Recognizing text with perspective distortion in natural scenes. 2013 IEEE International Conference on Computer Vision, pp. 569–576. IEEE, Piscataway (2013)
37. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. Int. J. Comput. Vision 116(1), 1–20 (2016)
38. Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. Expert Syst. Appl. 41, 8027–8048 (2014)
39. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.J.: COCO-Text: Dataset and benchmark for text detection and recognition in natural images. CoRR, abs/1601.07140 (2016)
40. Nayef, N., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., 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. vol. 1, pp. 1454–1459. IEEE, Piscataway (2017)
41. Shi, B., Yao, C., Liao, M., Yang, M., Xu, P., Cui, L., et al.: ICDAR2017 competition on reading chinese text in the wild (RCTW-17). In: Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on, vol. 1, pp. 1429–1434. IEEE, Piscataway (2017)
42. Ch'ng, C.K., Chan, C.S., Liu, C.: Total-Text: Towards orientation robustness in scene text detection. Int. J. Doc. Anal. Recogn. (IJDAR) 23, 31–52 (2020)
43. Pan, Y., Hou, X., Liu, C.: A hybrid approach to detect and localize texts in natural scene images. IEEE Trans. Image Process. 20(3), 800–813 (2011)
44. Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., et al.: ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition–RRC-MLT-2019. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1582–1587. IEEE, Piscataway (2019)
45. Borisyuk, F., Gordo, A., Sivakumar, V.: 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. KDD '18, p. 71–79. Association for Computing Machinery, New York (2018)
46. Wolf, C., Jolion, J.M.: Object count/area graphs for the evaluation of object detection and segmentation algorithms. Int. J. Document Anal. Recogn. (IJDAR) 8(4), 280–296 (2006)
47. Sabir, A., Moreno Noguer, F., Padró, L.: Textual visual semantic dataset for text spotting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 542–543. IEEE, Piscataway (2020)
48. Zharikov, I., Nikitin, P., Vasiliev, I., Dokholyan, V.: DDI-100: Dataset for text detection and recognition. In: Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control, pp. 1–5. ACM, New York (2020)
49. Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., Yan, J.: FOTS: Fast oriented text spotting with a unified network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5676–5685. IEEE, Piscataway (2018)
50. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9365–9374. IEEE, Piscataway (2019)
51. Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 11474–11481. AAAI Press, Palo Alto (2020)
52. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. Int. J. Comput. Vision 111(1), 98–136 (2015)
53. Santosh, K.C., Belaïd, A.: Document information extraction and its evaluation based on client's relevance. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 35–39. IEEE Computer Society, Los Alamitos (2013)
54. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Deep structured output learning for unconstrained text recognition. Paper presented at 3rd international conference on learning representations, San Diego, CA, 7–9 May 2015
55. Lee, C.Y., Osindero, S.: Recursive recurrent nets with attention modeling for OCR in the wild. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2231–2239. IEEE, Piscataway (2016)
56. Cohen, W., Ravikumar, P., Fienberg, S.: A comparison of string metrics for matching names and records. In: KDD Workshop on Data Cleaning and Object Consolidation. vol. 3, pp. 73–78. AAAI Press, Palo Alto (2003)
57. Weinman, J.J., Learned-Miller, E., Hanson, A.R.: Scene text recognition using similarity and a lexicon with sparse belief propagation. IEEE Trans. Pattern Anal. Mach. Intell. 31(10), 1733–1746 (2009)
58. Santosh, K.C.: G-DICE: Graph mining-based document information content exploitation. Int. J. Doc. Anal. Recogn. (IJDAR) 18(4), 337–355 (2015)
59. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2963–2970. IEEE, Piscataway (2010)
60. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. 22(10), 761–767 (2004)

61. Wu, D., Wang, R., Tian, X., Liang, D., Cao, X.: The keywords spotting with context for multi-oriented chinese scene text. In: 2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM), pp. 1–5. IEEE, Piscataway (2018)
62. Yang, Q., Cheng, M., Zhou, W., Chen, Y., Qiu, M., Lin, W., et al.: Incep-Text: A new inception-text module with deformable PSROI pooling for multi-oriented scene text detection. In: IJCAI, pp. 1071–1077. Morgan Kaufman, San Mateo (2018)
63. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9. IEEE, Piscataway (2015)
64. Chaves, D., Saikia, S., Fernández Robles, L., Alegre, E., Trujillo, M.: Una revisión sistemática de métodos para localizar automáticamente objetos en imágenes. Revista Iberoamericana de Automática e Informática industrial 15(3), 231–242 (2018)
65. Ye, J., Chen, Z., Liu, J., Du, B.: TextFuseNet: Scene text detection with richer fused features. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pp. 516–522. International Joint Conferences on Artificial Intelligence Organization (2020)
66. Zhang, S.X., Zhu, X., Hou, J.B., Liu, C., Yang, C., Wang, H., et al.: Deep relational reasoning graph network for arbitrary shape text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9699–9708. IEEE, Piscataway (2020)
67. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., et al.: EAST: An efficient and accurate scene text detector. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2642–2651. IEEE, Piscataway (2017)
68. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440. IEEE, Piscataway (2015)
69. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: TextBoxes: A fast text detector with a single deep neural network. In: AAAI, pp. 4161–4167. AAAI Press, Palo Alto (2017)
70. Tian, S., Lu, S., Li, C.: WeText: Scene text detection under weak supervision. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1501–1509. IEEE, Piscataway (2017)
71. Deng, D., Liu, H., Li, X., Cai, D.: PixelLink: Detecting scene text via instance segmentation. In: AAAI, pp. 6773–6780. AAAI Press, Palo Alto (2018)
72. Tian, Z., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: European Conference on Computer Vision, pp. 56–72. Springer, Berlin (2016)
73. He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., Li, X.: Single shot text detector with regional attention. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 3066–3074. IEEE, Piscataway (2017)
74. Hu, H., Zhang, C., Luo, Y., Wang, Y., Han, J., Ding, E.: WordSup: Exploiting word annotations for character based text detection. In: ICCV, pp. 4950–4959. IEEE, Piscataway (2017)
75. Jiang, Y., Zhu, X., Wang, X., Yang, S., Li, W., Wang, H., et al.: R2CNN: Rotational region cnn for orientation robust scene text detection. CoRR, abs/1706.09579 (2017)
76. Li, H., Wang, P., Shen, C.: Towards end-to-end text spotting with convolutional recurrent neural networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5238–5246. IEEE, Piscataway (2017)
77. Qin, S., Manduchi, R.: Cascaded segmentation-detection networks for word-level text spotting. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 01, pp. 1275–1282. IEEE Computer Society, Los Alamitos (2017)
78. He, D., Yang, X., Liang, C., Zhou, Z., Alexander, G., Ororbia, I., et al.: Multi-scale FCN with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In: CVPR, pp. 474–483. IEEE, Piscataway (2017)
79. Van, D.N., Lu, S., Bai, X., Ouarti, N., Mokhtari, M.: Max-pooling based scene text proposal for scene text detection. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp. 1295–1300. IEEE, Piscataway (2017)
80. Dai, Y., Huang, Z., Gao, Y., Xu, Y., Chen, K., Guo, J., et al.: Fused text segmentation networks for multi-oriented scene text detection. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3604–3609. IEEE, Piscataway (2018)
81. Wang, W., Xie, E., Li, X., Liu, X., Liang, D., Zhibo, Y., et al.: PAN++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. IEEE Trans. Pattern Anal. Mach. Intell. (2021)
82. Zhou, Y., Liu, S., Zhang, Y., Wang, Y., Lin, W.: Perspective scene text recognition with feature compression and ranking. In: Computer Vision - ACCV 2014 Workshops, pp. 181–195. Springer International Publishing, Cham (2015)
83. Lu, S., Chen, B.M., Ko, C.C.: A partition approach for the restoration of camera images of planar and curled document. Image Vis. Comput. 24(8), 837–848 (2006)
84. Prasad, S., Kong, A.W.K.: Using object information for spotting text. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018, pp. 559–576. Springer International Publishing, Cham (2018)
85. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149 (2015)
86. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788. IEEE, Piscataway (2016)
87. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376. ACM, New York (2006)
88. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings. ICML, San Diego (2015)
89. Litman, R., Anschel, O., Tsiper, S., Litman, R., Mazor, S., Manmatha, R.: SCATTER: Selective context attentional scene text recognizer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11962–11972. IEEE, Piscataway (2020)
90. Shi, B., Bai, X., Yao, C.: 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(11), 2298–2304 (2016)
91. Xie, Z., Huang, Y., Zhu, Y., Jin, L., Liu, Y., Xie, L.: Aggregation cross-entropy for sequence recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6538–6547. IEEE, Piscataway (2019)
92. Yue, X., Kuang, Z., Lin, C., Sun, H., Zhang, W.: RobustScanner: Dynamically enhancing positional clues for robust text recognition. In: European Conference on Computer Vision, pp. 135–151. Springer, Berlin (2020)
93. Yan, R., Peng, L., Xiao, S., Yao, G.: Primitive representation learning for scene text recognition. In: CVPR, pp. 284–293. IEEE, Piscataway (2021)
94. Zheng, T., Chen, Z., Fang, S., Xie, H., Jiang, Y.G.: CDistNet: Perceiving multi-domain character distance for robust text recognition. arXiv preprint arXiv:211111011 (2021)
95. Bhunia, A.K., Sain, A., Kumar, A., Ghose, S., Chowdhury, P.N., Song, Y.Z.: Joint visual semantic reasoning: Multi-stage decoder for text recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14940–14949. IEEE, Piscataway (2021)
96. He, Y., Chen, C., Zhang, J., Liu, J., He, F., Wang, C., et al.: Visual Semantics Allow for Textual Reasoning Better in Scene Text Recognition. arXiv preprint arXiv:211212916 (2021)
97. Bartz, C., Yang, H., Meinel, C.: STN-OCR: A single neural network for text detection and text recognition. CoRR, abs/1707.08831 (2017)
98. Cheng, Z., Xu, Y., Bai, F., Niu, Y., Pu, S., Zhou, S.: Aon: Towards arbitrarily-oriented text recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5571–5579. IEEE, Piscataway (2018)

99. Li, H., Wang, P., Shen, C., Zhang, G.: Show, attend and read: A simple and strong baseline for irregular text recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33(01), pp. 8610–8617. AAAI Press, Palo Alto (2019)
100. Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7098–7107. IEEE, Piscataway (2021)
101. Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L.: ABCNet: Real-time scene text spotting with adaptive bezier-curve network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9809–9818. IEEE, Piscataway (2020)
102. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556 (2014)
103. Kim, K.H., Cheon, Y., Hong, S., Roh, B.S., Park, M.: PVANET: Deep but lightweight neural networks for real-time object detection. CoRR, abs/1608.08021 (2016)
104. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. Paper presented at international conference on machine learning, Long Beach, CA, 10–15 June 2019, pp. 6105–6114
105. Bušta, M., Neumann, L., Matas, J.: Deep TextSpotter: An end-to-end trainable scene text localization and recognition framework. In: Computer Vision (ICCV), 2017 IEEE International Conference on, pp. 2223–2231. IEEE, Piscataway (2017)
106. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525. IEEE, Piscataway (2017)
107. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. 115(3), 211–252 (2015)
108. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al.: TensorFlow: A system for large-scale machine learning. In: OSDI'16: Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, vol. 16, pp. 265–283. ACM, New York (2016)
109. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678. ACM, New York (2014)
110. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al.: PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32, pp. 8024–8035. Curran Associates, Inc., Red Hook (2019)

How to cite this article: Blanco-Medina, P., Fidalgo, E., Alegre, E., González-Castro, V.: A survey on methods, datasets and implementations for scene text spotting. IET Image Process. 16, 3426–3445 (2022). https://doi.org/10.1049/ipr2.12574