Text Detection and Recognition in Natural Scene, Entering in the Deep Learning World

Abner Turkieltaub (92286558)
Department of Computer Science
The University of British Columbia
abner7@cs.ubc.ca

Alireza Iranpour (94447984)
Department of Computer Science
The University of British Columbia
Iranp01@cs.ubc.ca

Shivam Thukral (94064177)
Department of Computer Science
The University of British Columbia
tshivam2@cs.ubc.ca

Abstract

The task of detecting and recognizing text from a real world image has received a lot of attention from the computer vision community and also from the machine learning community. This problem is not only very challenging, but it is also very important for its numerous potential applications from converting handwritten notes into text documents to industrial automation. This survey serves to provide an introduction to the topic and shed light on the latest advances brought about by employing deep learning methods.

1 Introduction

Text is one of the most expressive means of communication. Such texts can be found in a variety of places ranging from streets, walls, documents, etc. Also, the increase in availability of high performance mobile devices make image acquisition and processing possible anytime and anywhere. Considering these factors, text detection and recognition becomes an important research area. Now, with the advancement of pattern recognition, computer vision, and machine learning techniques, we can solve more complex problems dealing with text localisation and recognition. While some researchers view optical character recognition (OCR) as a solved problem, text detection and recognition in natural scenes still has many hurdles to cross before we can fully accept such statement.

This paper presents a comprehensive survey of text detection, tracking and recognition methods with special focus on recent technological advancements. This survey highlights state-of-the-art techniques before the deep learning era and after its boom.

1.1 Problem Definition

Our problem can be divided into three broad categories:

- **Scene Text Detection** : is process of predicting the presence of text and localisation of each instance (if any), at a word or line level from natural scenes.
- **Scene Text Recognition** : The process of converting detected text regions into readable forms either by computers or humans.
- **End-to-End Systems** : This involves combining the above mentioned two steps into one. In this, we will localize and recognize the text at the same time.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

1.2 Challenges

Detection and recognition of text in the wild comes with many challenges:

1. **Variability and Diversity** : Texts that appear in natural scenes are of high variability and diversity. For instance, they can be found in various languages, colors, fonts, sizes, shapes, positions, etc.
2. **Interference of complex backgrounds** : Unpredictable patterns and scenes that make up the backgrounds pose serious challenges in detection and recognition. Cases of similarity between the text and its background or partial obstructions caused by foreign objects can result in confusion and error.
3. **Poor Imaging conditions** : The quality of text images that have been captured in the wild can never be guaranteed. These images can be of low resolution or subject to distortion and blurriness due to inappropriate shooting conditions such as distance, angle or focus.

The rest of the paper is outlined as follows: Section 2, highlights some of the most successful methods before deep learning. In section 3, we list and summarize the algorithms based on deep learning approaches. This survey also draws comparisons between the various methods. In section 4, we explain the auxiliary techniques which made deep learning methods so successful in performing text detection and recognition. Section 5 and 6, summarize and describe the various datasets and evaluation protocols. In section 7, we do a comparison between approaches followed before and after deep learning era. Finally, section 8 presents applications of scene text localization and recognition, and section 9 tries to predict the future trends in this field.

2 Conventional Methods

In this section some of the more recent and successful algorithms prior to the deep learning era are presented. We divide them into three subsections according to the task they perform.

2.1 Detection

We can divide the traditional methods for scene text detection in three categories: Sliding Window (SW) based methods, Connected Component (CC) based methods and hybrid methods. Sliding window based methods, as the name suggests, search for text in the image by scanning different fragments (or windows) with different sizes and in different coordinates and then use machine learning techniques to identify texts. These methods are generally slow since multiple windows of different sizes must be processed. Connected component based methods extract character candidates from images by connected component analysis followed by grouping character candidates into text. Inside the connected component based category, we have Maximally Stable Extremal Region (MSER) based methods, this methods extract character candidates using MSER, assuming similar color within each character. The advantages of this method are that it is robust, fast to compute and independent of scale. But, this method also have some issues, it detects a lot of false positives so it needs some posterior pruning process and also needs to merge character candidates to detect words. The MSER approach has been used in several works, for example in [15] it is used to detect license plates in natural images, in [21] is used in more general images for end-to-end systems and in [10] this method is combined with convolutional neural networks. Another important representative of connected component based methods is the Stroke Width Transform (SWT) introduced by Epshtein et al. [5], instead of assuming similar color within a character, this method assumes consistent stroke width within each character. Figure 1 illustrate MSER and SWT methods.

2.2 Recognition

A basic idea for this task is to first identify individual characters, and then recognize each of them individually. Here, we briefly describe two methods used to solve this problem.

2.2.1 Top-Down and Bottom-Up Cues

In this approach by Mishra et al. [20], bottom-up cues are extracted by detecting each character individually. For this purpose, a Conditional Random Field model is employed to identify the

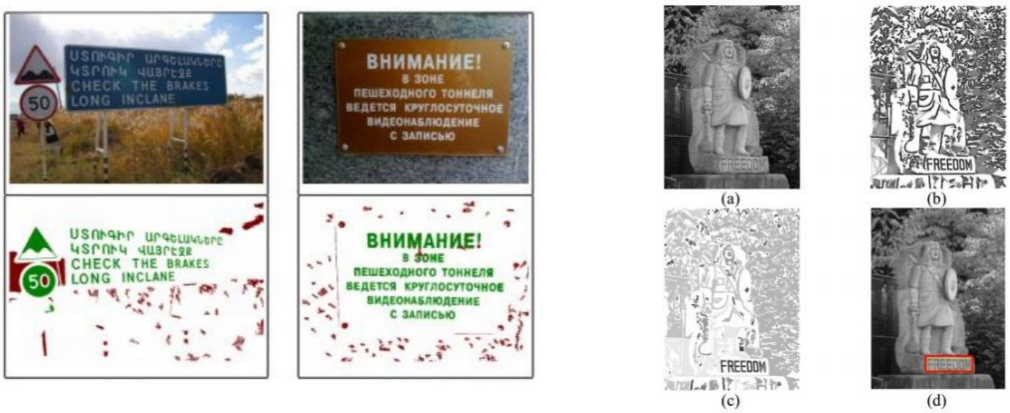

Figure 1: The left image [21] shows an example with character candidates (in green) after using MSER. The right image [5] illustrate the SWT process: SWT converts the image (a) from containing gray values to an array containing likely stroke widths for each pixel (b). This information suffices for extracting the text by measuring the width variance in each component as shown in (c) because text tends to maintain fixed stroke width. This puts it apart from other image elements such as foliage. The detected text is shown in (d).

locations of true characters as well as the words they represent as a whole. The detection of characters is performed using approaches based on sliding windows where multiple windows of different scales and spatial positions are considered to overcome the challenges of intra-character and inter-character confusions. Pruning methods are then used to remove the least predictable character windows based on classifier confidence. The bottom-up approach may not be able to discard all of the false positives. As a result, lexicon-based top-down cues such as language statistics are also utilized to remove the rest of the false positives as well as support recognition. Recognition of the text is performed by optimizing the energy function associated with the random field model.

2.2.2 Tree-Structured Model

As opposed to the sliding window approach which does not take into account the global structure, the tree-structured model by Shi et al. [28] makes use of part-based tree structure to perform detection and recognition at the same time by modeling each type of character. Once the potential locations of characters have been identified, based on the detection results, a CRF model is built on the potential locations where final recognition of each word is done by minimizing a cost function defined on the random field.

2.3 End-to-End Systems

The third category of systems, called end-to-end systems, performs text detection and recognition together. The main advantage of such systems is that they can share features between these two tasks and at the same time share the computational burden.
Wang at el. [30] proposed a method in which each word is represented as a special kind of object and the characters comprising the word as a part of the object. They introduced the idea of "word spotting", which is a holistic matching technique. In this, character and word models are used to match specific words in a given lexicon with image patches. Here, character models are trained with histogram of oriented gradients (HOG) features and a random ferns classifier is also used. They obtain character responses by multi-scale sliding window classification, and non-maximal suppression to localize the character candidates. Then these are grouped into words through Pictorial Structure that inputs scores and locations of characters to determine optimal configuration of the word using small lexicon.

Figure 2: Selected results of end-to-end methods on the ICDAR dataset

3 Deep Learning Era

In this section, we will present methods for detection, recognition and end-to-end systems that use deep learning techniques. We divide the section into three parts, each taking care of one specific task.

3.1 Detection

Due to the great success of Deep Convolutional Neural Network (CNN) in generic object detection, scene text detection has also been greatly improved by regarding text lines as objects. Many approaches combined classical methods to find text candidates and deep learning to filter those candidates. But in recent years, many methods ([36], [25], [7], [33]) have been developed that use deep learning not only to select the final candidates, but also to detect the whole list of candidates. Here, we briefly introduce two of them.

3.1.1 EAST: Efficient and Accurate Scene Text Detector

This method was proposed by Zhou et al. [36], it is a very clean method in the sense that unnecessary steps are avoided. Almost all the detection work is done by a fully convolutional network (FCN). The FCN selects regions of the image as text candidates and then a simple Non-Maximum Suppression (NMS) algorithm decides the final text regions. The NMS algorithm does the following: consider C as the set of candidates from the FCN, and A as the set of accepted candidates (initially empty). While C is non empty, select the higher score candidate in C, include it in A and then, compute the intersection over union area of this candidate with all remaining candidates in C, eliminating those above some fix threshold.

3.1.2 SegLink

Presented by Shi et al. [25], the main idea of this method is to decompose text into two locally detectable elements that the authors called segments and links (See Figure 3). A segment is an oriented box covering a part of a word or a text line; A link connects two adjacent segments, indicating that they belong to the same word or text line. Both elements are detected by a fully-convolutional neural network. Final predictions are obtained by combining segments connected by links. SegLink is able to detect long lines of Latin and non-Latin text (such as Chinese) in different orientations.

3.2 Recognition

3.2.1 Connectionist Temporal Classification (CTC)

In CTC-based methods, the conditional probability $\Pr(L|y)$ is computed where L and y represent the label sequence and the pre-frame prediction of RNN, respectively. CRNN was proposed by Shi et

Figure 3: In this image [25], we can see the segments (yellow boxes), links (green lines), and the final output (green boxes).

al. [26] where CNN and RNN were stacked together for text recognition. Their proposed model comprised of three parts: (1) convolutional layers used for feature extraction, (2) recurrent layers used for predicting label distributions, and (3) transaction layer (CTC) which translates pre-frame predictions to the final label predictions. Gao et al. [6] used stacked convolutional layers in place of RNN to more effectively capture the contextual dependencies. Their proposed approach had the advantages of lower computational cost as well as facilitated parallel computation.

3.2.2 Attention-based methods

The attention mechanism was initially employed to improve neural machine translation systems. Lee et al. [4] proposed a recursive recurrent neural network with attention modeling for lexicon-free text recognition. The advantages of their method included (1) using recursive CNNs which allowed for effective and efficient feature extraction, (2) learning character-level language model which obviated the need for n-grams, and (3) using a soft attention mechanism which allowed for selective exploitation of image features in a coordinated manner. Cheng et al. [4] proposed Focusing Attention Network (FAN) to tackle the problem of attention drift in the existing methods where they failed to get accurate alignments between feature areas and the target in complicated or low-quality images. Bai et al. [1] came up with a metric called edit probability (EP) which alleviates the effect of misalignments between the ground truth and the attention's output by focusing on the missing, superfluous and unrecognized characters in the training process. Liu et al. [16] proposed a binary convolutional encoder-decoder network (B-CEDNet) for natural scene text processing (NSTP) where computational cost is reduced by training the decoder part under binary constraints.

3.3 End-to-End Systems

Text spotting systems are build to do the task of detection and recognition end-to-end systems. Here two discuss two major works in this area:

3.3.1 Using CNN

Jaberberg et al. [12]: trained a CNN which predicts a text present/absent score given an 24×24 input image. From this score they developed a Saliency Map by scanning over 16 scales. Next, horizontal bounding box proposals are detected by aggregating the output of standard edge boxes and aggregate channel feature detectors. Each box is filtered to remove false positives by random forests and its position and location is further refined by a CNN regressor.

3.3.2 DeepText

This algorithm [2] proposed an end-to-end trainable scene text localization and recognition framework. The detection branch adopted YOLOv2 and the text recognition branch in which text proposals are mapped into fixed height tensor by bi-linear sampling and then CTC-based regularization is used to transform this into strings. This word spotter can solve word spotting in multi-orientation problem.

However, this method does not have shareable features, which means there is no relation between text detection and recognition modules.

4 Additional Techniques

The success of above mentioned methods will not be fully justified if we do not talk about some of the supplementary tasks which aided deep learning techniques. We summarize these techniques as follows:

4.1 Synthetic Data

The success of deep learning models highly depends on the data. Considering the fact that the current human-annotated datasets are small, this problem becomes more important.
Jaderberg et al. [11] produces synthetic text data by first changing the font, orientation and color of the text and then placing it on randomly cropped images. By doing so and training on this dataset we can achieve very good performance results. Zhan et al[35] performs text synthesis with deep learning techniques in a realistic manner by doing semantic segmentation. Semantic segmentation groups clusters as semantic clusters. By doing so text was only placed at sensible places (walls,tables etc.) and not randomly as in previous methods.

4.2 Bootstrapping

This is divided into word and character level annotations. They differ in the way how much filtering is done.

- **Word Level**: Rong et al [23] trained a FCN, which predicts if a pixel from an image belongs to text or not. Next, where we have high confidence we extract features using MSER. Single Linkage Criterion is used to decide the final prediction.
- **Character Level**: is better than word-level annotations. WeText[29], performs filtering by thresholding to select the most reliable prediction candidates and filtering is done by word-level annotations. WordSup [9] is a scoring based method, which uses eigenvalue based metric on a co-variance matrix computed by the center of each selected character boxes.

4.3 De-blurring

Both text-detection and recognition is sensitive to deblurring. Hradis et al[8] collected a dataset of images which was taken properly. In their deblurring method the FCN maps the deblurred image to a deblurred image by convoluting it against kernels that perform de-focus.

5 Datasets

Many datasets have been and are being created to measure the accuracy of different methods. This datasets differ not only in their sizes, but also in the challenges they address. In this section, we list and briefly introduce some of them.

- Many widely used datasets come from the ICDAR Robust Reading Competition ([18], [17], [24], [14], [13] and [27]). This is a big competition that has been held seven times, in 2003, 2005, 2011, 2013, 2015, 2017 and 2019. In the first versions, the datasets consisted only in images reasonably well focused, with text in English and horizontally oriented. In later versions more challenges have been added, including arbitrary oriented text, multi-lingual text, and text in videos among others. For comparison between methods we will use the ICDAR 2013 dataset containing 229 train examples and 233 test examples, with horizontal, English text.
- The Chinese Text in the Wild (CTW) [34] dataset contains 32,285 high resolution street view images, with Chinese text.

- Total-Text [3] has a total of 1,555 images, and includes a large proportion of curved and multi-oriented text.
- The Street View Text (SVT) [31], [30] dataset is a collection of street view images taken from Google street view.
- The MSRA-TD500 [32] contains multi-oriented text in English and Chinese. It is only for detection.
- CASIA-10K is a Chinese scene text dataset from last year. It contains 10,000 images, 7,000 images are for training and 3,000 images are for testing. For each text line, 8 coordinates of a quadrilateral are annotated. It is only for detection.
- The IIIT 5K-Word [19] contains 5,000 cropped word images from the real world and born-digital images. It is only for recognition.
- The SVT-Perspective (SVTP) [22] contains images picked from the side-view images in Google Street View. Many of them are heavily distorted by the non-frontal view angle. It was proposed for evaluating the performance of recognizing perspective text.

6 Evaluation Protocols

For the tasks of detection and recognition, there are different ways to evaluate the performance of any given model. The objective of this section is to present the general ideas behind the metrics used for evaluating algorithms in this tasks.

6.1 Evaluation Protocols for Text Detection

For detection, each image have a set of ground truth rectangles (in some cases, more general polygons could be used), that enclose the text in the images. The algorithms for this task must define their own rectangles to enclose the text and the idea is that the rectangles defined by the algorithms should be in same sense close to the ground truth rectangles. There are basically three metrics to measure the performance in this task, and they are:

- Precision (P): is the proportion of predicted text instances that can be matched to ground truth labels. So good precision means few false positives.
- Recall (R): is the proportion of ground truth labels that have correspondents in the predicted list. So good recall means a lot of true positives.
- F1-score (F_1): this metric correspond to the harmonic mean of precision and recall, that is: $F_1 = 2\frac{PR}{P+R}$. Basically a good F1-score implies good precision and recall.

To actually compute these performance indicators, the list of predicted text instances should first be matched to the ground truth labels. To do this match and calculate the final scores, there exists different protocols. It is not the purpose of this work to go deeper into this protocols, so we will cut this discussion here.

6.2 Evaluation Protocols for Text Recognition and End-to-end

In text recognition, images contains only one word, so we just need to compare the predicted text with the real text in the image. The performance evaluation could be either character-level recognition rate (i.e. how many characters are recognized) or word level (whether the predicted word is completely correct). The ICDAR competition also introduces an edit-distance based performance evaluation. For end-to-end evaluation, we need to first match predicted and truth words in a similar way to that of text detection and then measure similarity with one of the methods used for text recognition.

7 Comparisons

To compare some methods used before and after deep learning era we report precision, recall and F_1-score in Table 1. We have to used different datasets since we did not find a common dataset to compare this techniques, but we select two ICDAR dataset from different years, but in the same

Algorithm	DataSet	Precision	Recall	F1 score
Neumann et al.	ICDAR 2003	0.65	0.64	0.64
Epshtein et al.	ICDAR 2003	0.73	0.60	0.66
EAST	ICDAR 2013	0.93	0.83	0.87
SegLink	ICDAR 2013	0.88	0.83	0.85

Table 1: Comparison of different detection techniques before and after deep learning

category, so the difficulty is not drastically different. As we can see in the Table 1, there has been major improvements in precision, recall and F_1-score. The complete credit for this improvement can be given to state-of-art algorithms which involve deep learning techniques.

8 Applications

In the past two decades, text detection and recognition in natural scenes has become an active research topic. Considering its importance, many models, algorithms and systems have been developed. Here, we briefly describe major applications of OCR:

1. **Industrial Automation** : Recognizing text on packages, houses, streets has a broad use in the industrial automation. In the case of autonomous vehicles, we need to detect and recognize text on panels, which give us information regarding geo-location, navigation, etc. Researchers form GRASP laboratory in University of Pennsylvania have developed a robot called "Graspy" which can read text characters in the wild.
2. **Multimedia Retrieval** : Recognizing text and extracting it from multimedia resources like images and videos enhances multimedia retrieval. This can be used to electronically store historical documents. It can also be used for automatic data entry. Some companies are already using this technology e.g. SF-Express. Also, extracted text from videos in the form of subtitles can be used in automatic content tagging and recommender systems. This can also be used to perform user sentiment analysis.
3. **Visual Input and Access** : Automatic sign recognition and translation systems enable users to overcome language barriers. For example, Google Goggle APP can perform the task of instant translation which is extremely useful and time saving for people who travel or read documents in foreign languages. Also, developing text-to-speech devices can assist visually impaired and illiterate people in understanding. The University of Maryland has developed a text recognition system for this purpose.

9 Future Trends

Over the past decade, we have seen considerable progress in this field especially as a result of the deep learning boom. However, there are plenty of challenging problems that still need to be addressed. This area has undergone a lot of development, to an extent that Google Inc. and Amazon Inc. are providing their own APIs for text detection and recognition.

1. **Better Data** : Most of the currently available datasets provide annotations in the form of bounding box and written text. Detailed information like WordArt, occlusions can be of much more use. Hence, apart from datasets capturing real-world challenges, they should be more informative as well.
2. **Multi-Language** : Most of current methods just deal with the English language. There has been some work in other languages as well (for example, Korean, Chinese etc.). Considering that we have thousands of languages, building a unified model that can incorporate most of the languages can be of great academic value. One approach to solve this problem is to find common patterns between languages and then train models on them.
3. **Synthetic Data** : By using synthetic data we can simulate different conditions such as different text orientations, occlusions, etc. It still remains a challenging task to synthesize diverse and realistic text data which can capture the majority of the variations.

4. **Generalisation** : Many algorithms fails when it comes to generalization. If we train on one dataset and test on another, the models perform poorly. This is important as we want to incorporate as many cases as possible. One solution to this problem is to pool all the existing datasets together and train the model on that.
5. **Efficiency** : Such deep learning systems have hardware limitations. Small devices like tablets and mobile phones cannot run such methods without GPU's. One direction to solve this problem is to do model compression and build lightweight models which can run on less computationally powerful machines,
6. **Robustness of Models** : Although current approaches in text recognition give promising results, in some cases, they fail at detection. In such cases, the model itself is not robust enough. We feel that this is mainly due to the internal operating mechanisms of deep neural networks.

10 Conclusions

Detecting and recognising text in natural scenes is an important task due to its wide application domain. From this literature survey, we have tried to summarise the major approaches which were followed both before and after the deep learning boom. We have seen how deep learning model based approaches have replaced manual search and design for pattern and features. With the improvement of such models and techniques the focus is now shifting towards more complex problems in this domain.

References

[1] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou. "Edit probability for scene text recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pages 1508–1516 (cited on page 5).

[2] M. Busta, L. Neumann, and J. Matas. "Deep textspotter: An end-to-end trainable scene text localization and recognition framework". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pages 2204–2212 (cited on page 5).

[3] C. K. Ch'ng and C. S. Chan. "Total-text: A comprehensive dataset for scene text detection and recognition". In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Volume 1. IEEE. 2017, pages 935–942 (cited on page 7).

[4] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou. "Focusing attention: Towards accurate text recognition in natural images". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pages 5076–5084 (cited on page 5).

[5] B. Epshtein, E. Ofek, and Y. Wexler. "Detecting text in natural scenes with stroke width transform". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pages 2963–2970 (cited on pages 2, 3).

[6] Y. Gao, Y. Chen, J. Wang, and H. Lu. "Reading scene text with attention convolutional sequence modeling". In: *arXiv preprint arXiv:1709.04303* (2017) (cited on page 5).

[7] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu. "Deep direct regression for multi-oriented scene text detection". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pages 745–753 (cited on page 4).

[8] M. Hradiš, J. Kotera, P. Zemcık, and F. Šroubek. "Convolutional neural networks for direct text deblurring". In: *Proceedings of BMVC*. Volume 10. 2015, page 2 (cited on page 6).

[9] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding. "Wordsup: Exploiting word annotations for character based text detection". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pages 4940–4949 (cited on page 6).

[10] W. Huang, Y. Qiao, and X. Tang. "Robust scene text detection with convolution neural network induced mser trees". In: *European Conference on Computer Vision*. Springer. 2014, pages 497–511 (cited on page 2).

[11] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. "Synthetic data and artificial neural networks for natural scene text recognition". In: *arXiv preprint arXiv:1406.2227* (2014) (cited on page 6).

[12] M. Jaderberg, A. Vedaldi, and A. Zisserman. “Deep features for text spotting”. In: *European conference on computer vision*. Springer. 2014, pages 512–528 (cited on page 5).

[13] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. “ICDAR 2015 competition on robust reading”. In: *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE. 2015, pages 1156–1160 (cited on page 6).

[14] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras. “ICDAR 2013 robust reading competition”. In: *2013 12th International Conference on Document Analysis and Recognition*. IEEE. 2013, pages 1484–1493 (cited on page 6).

[15] H. W. Lim and Y. H. Tay. “Detection of license plate characters in natural scene with MSER and SIFT unigram classifier”. In: *2010 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*. IEEE. 2010, pages 95–98 (cited on page 2).

[16] Z. Liu, Y. Li, F. Ren, W. L. Goh, and H. Yu. “Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network”. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018 (cited on page 5).

[17] S. M. Lucas. “ICDAR 2005 text locating competition results”. In: *Eighth International Conference on Document Analysis and Recognition (ICDAR’05)*. IEEE. 2005, pages 80–84 (cited on page 6).

[18] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. “ICDAR 2003 robust reading competitions”. In: *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* Citeseer. 2003, pages 682–687 (cited on page 6).

[19] A. Mishra, K. Alahari, and C. Jawahar. “Scene text recognition using higher order language priors”. In: 2012 (cited on page 7).

[20] A. Mishra, K. Alahari, and C. Jawahar. “Top-down and bottom-up cues for scene text recognition”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pages 2687–2694 (cited on page 2).

[21] L. Neumann and J. Matas. “A method for text localization and recognition in real-world images”. In: *Asian Conference on Computer Vision*. Springer. 2010, pages 770–783 (cited on pages 2, 3).

[22] T. Quy Phan, P. Shivakumara, S. Tian, and C. Lim Tan. “Recognizing text with perspective distortion in natural scenes”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pages 569–576 (cited on page 7).

[23] L. Rong, E. MengYi, L. JianQiang, and Z. HaiBin. “Weakly Supervised Text Attention Network for Generating Text Proposals in Scene Images”. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Volume 1. IEEE. 2017, pages 324–330 (cited on page 6).

[24] A. Shahab, F. Shafait, and A. Dengel. “ICDAR 2011 robust reading competition challenge 2: Reading text in scene images”. In: *2011 international conference on document analysis and recognition*. IEEE. 2011, pages 1491–1496 (cited on page 6).

[25] B. Shi, X. Bai, and S. Belongie. “Detecting oriented text in natural images by linking segments”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pages 2550–2558 (cited on pages 4, 5).

[26] B. Shi, X. Bai, and C. Yao. “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.11 (2016), pages 2298–2304 (cited on page 5).

[27] B. Shi, C. Yao, M. Liao, M. Yang, P. Xu, L. Cui, S. Belongie, S. Lu, and X. Bai. “Icdar2017 competition on reading chinese text in the wild (rctw-17)”. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Volume 1. IEEE. 2017, pages 1429–1434 (cited on page 6).

[28] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang. “Scene text recognition using part-based tree-structured character detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pages 2961–2968 (cited on page 3).

[29] S. Tian, S. Lu, and C. Li. “Wetext: Scene text detection under weak supervision”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pages 1492–1500 (cited on page 6).

[30] K. Wang, B. Babenko, and S. Belongie. “End-to-end scene text recognition”. In: *2011 International Conference on Computer Vision*. IEEE. 2011, pages 1457–1464 (cited on pages 3, 7).

[31] K. Wang and S. Belongie. “Word spotting in the wild”. In: *European Conference on Computer Vision*. Springer. 2010, pages 591–604 (cited on page 7).

[32] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. “Detecting texts of arbitrary orientations in natural images”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pages 1083–1090 (cited on page 7).

[33] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. “Scene text detection via holistic, multi-channel prediction”. In: *arXiv preprint arXiv:1606.09002* (2016) (cited on page 4).

[34] T.-L. Yuan, Z. Zhu, K. Xu, C.-J. Li, and S.-M. Hu. “Chinese text in the wild”. In: *arXiv preprint arXiv:1803.00085* (2018) (cited on page 6).

[35] F. Zhan, S. Lu, and C. Xue. “Verisimilar image synthesis for accurate detection and recognition of texts in scenes”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pages 249–266 (cited on page 6).

[36] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. “EAST: an efficient and accurate scene text detector”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pages 5551–5560 (cited on page 4).