An Overview of State-of-the-art Solutions for Scene Text Detection

Mladen Džida, Davor Vukadin, Marin Šilić, Goran Delač, Klemo Vladimir
University of Zagreb Faculty of Electrical Engineering and Computing, Zagreb, Croatia
mladen.dzida@fer.hr, davor.vukadin@fer.hr, marin.silic@fer.hr, goran.delac@fer.hr, klemo.vladimir@fer.hr

***Abstract*—Scene text detection is a task of identifying text regions and labeling them with bounding boxes in a complex background. It has received a lot of attention recently and has become far from unsolvable due to progress of deep learning for computer vision and also due to rapid development of computer hardware which is able to process complex neural networks. Some of the most common challenges that make this task difficult are irregular text shapes, text interferences, very complex background, different text sizes and low image quality. This paper presents an overview of state-of-the-art solutions for scene text detection where ICDAR 2015 was used as a benchmark dataset. We compare solutions with respect to precision, recall and F-score.**

***Keywords*—Scene text, Text detection, Bounding box, Multi-oriented text, Segmentation, Convolutional neural network**

I. INTRODUCTION

Today, textual information is everywhere around us and correctly interpreting it is crucial for utilizing the benefits that society has to offer. This paper focuses on research of textual information that can be found in an outdoor environment, i.e. scene text. Typically, these information are used for wayfinding, advertising, traffic control and displaying of art or any other type of information that is meant to be seen by large number of people. By using machine learning, we could extract the meaning from these texts and use it to help automate outdoor tasks. For example, text information can be used in training of self driving cars, drones and Text-to-Speech models for visually impaired individuals. To achieve this, the automation system would first have to get the image of the scene text, then detect the location of the text in an image and finally recognize the text. Detecting or localizing the text in an image is finding and labeling the correct location of the text. Text recognition is a process that converts an image of text into a machine-readable text format [1]. This paper will focus only on the solutions for the problem of localizing the text in an image.

Detecting scene text in an image faces many challenges and the following are the most common ones:

- **Irregular text shapes**. Text in an outdoor environment often appears in various orientations and shapes. Irregular text shapes can include text that is curved, slanted or written at an angle, which can cause problems for models that are designed to detect text in a specific orientation or shape.
- **Text interferences**. Text interference refers to various factors that can make it difficult to detect text in natural scenes. For example, other objects can cause occlusion of the text in the image or there can be low contrast between text and the background, making it difficult to distinguish them.
- **Complex background**. Scene text in an image often appears on cluttered and dynamic backgrounds, i.e. complex backgrounds. Complex backgrounds can include textures, patterns, or other elements that are similar to the text. While these backgrounds can cause already mentioned text interferences, they can also produce false positives and detect text in the background patterns.
- **Different text sizes**. Scene text comes in various sizes. It can be small like on signs and labels or it can be large like on billboards and buildings. Models that are trained to detect text at a specific size may not perform well when they are given text that is of different size than what they have seen during training.
- **Low image quality**. Low image quality refers to factors such as poor lighting, blurriness and noise. Poor lighting can cause unwanted shadows and reflections, blurriness can make it difficult to distinguish the text from the background and noise can cause text interferences.

To overcome these challenges, many approaches have been proposed, but those from deep learning era have produced the best results. It is notable to mention that there were methods before the rise of deep learning that tackled the problem of scene text detection and they mostly adopted the Connected Components Analysis (CCA) or Sliding Window (SW) based classification [2]. Solutions that adopt these methods will not be covered in this overview, since all state-of-the-art solutions build models on deep learning methodologies.

The remaining sections of this paper are arranged as follows: In Section 2, the general deep learning methodologies for scene text detection are briefly reviewed. In Section 3, state-of-the-art solutions are described. In Section 4, comparison of the state-of-the-art solutions on benchmark dataset is carried out. Finally, conclusion is set in section 5.

II. RELATED WORK

Deep learning has played an important role in the field of scene text detection by enabling the development of

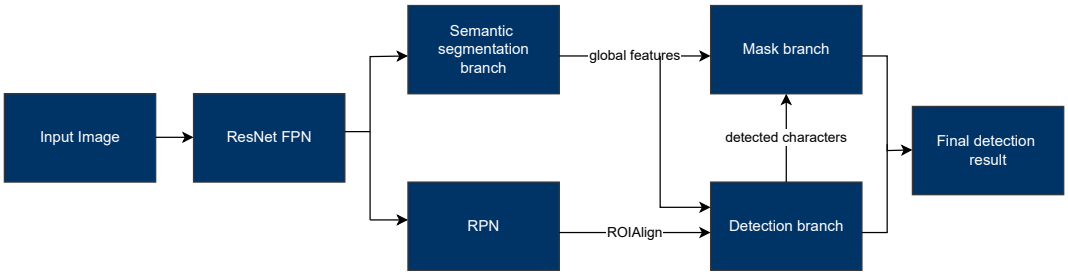

Fig. 1: TextFuseNet: pipeline

highly accurate and robust text detection models [2]. Before the deep learning era, feature extraction for scene text detection relied on hand-crafted features. These features were difficult to engineer and could not handle variations in text such as different fonts, orientations and scales. In contrast, deep learning models can automatically learn powerful representations from the data, allowing them to be more robust to variations in text. These models use convolutional neural networks (CNNs) to automatically learn features from images. Overall, the use of deep learning has led to significant improvements in the performance of scene text detection systems.

Recently, scene text detection experienced great improvement with the development of the CNNs. CNN-based algorithms for scene text detection can be classified into two categories: regression-based, segmentation based and hybrid methods [3]. These approaches will be explained in a more detail in the following subsections.

A. *Regression-based CNN methods*

Regression-based approaches [4]–[11] for scene text detection imply training a model to regress the coordinates of the bounding box for an object directly from the image [3]. Regression-based methods are typically simpler and faster than segmentation-based methods because they only need to predict a small number of bounding box coordinates, but they may not be as accurate when localizing occluded and small objects.

B. *Segmentation-based CNN methods*

Segmentation-based approaches [12]–[20] for scene text detection imply training a model for pixel-level prediction of objects, i.e. prediction of a segmentation mask for each object in the image [4]. Segmentation-based methods are typically more accurate than regression-based methods because they provide more detailed information about the object's shape and position in the image. However, these methods are usually more computationally expensive and require complex post-processing.

C. *Hybrid methods*

Hybrid approaches [21]–[24] combine feature maps produced by regression-based and segmentation-based methods into a sole feature representation which is used for scene text detection. The idea behind using both feature maps is reducing the negative effect of each of these approaches and having more useful information.

III. State-of-the-art approaches

In this section, state-of-the-art architectures for scene text detection will be described in detail. Each architecture will be described in a separate subsection.

TextFuseNet

TextFuseNet is a scene text detection framework that is using multi level feature extraction for detecting texts of arbitrary shapes [24]. The proposed method is a hybrid approach, i.e. the authors perform regression and segmentation. To help explain the reasoning behind such approach, they categorize all of the work in this area into two classes: character-based and word-based. The former detects characters and groups them into words, while the latter immediately detects words. The key idea of their work was extracting three semantically different types of features from the image and combining their information to detect text in various shapes, orientations and sizes. The three levels of features they extract are: character, word and global.

To extract features and later detect bounding boxes, the authors construct an architecture consisting of five parts: feature pyramid network (FPN), region proposal network (RPN), semantic segmentation branch, detection branch and mask branch. ResNet FPN is used as the backbone to extract feature maps of different sizes which they combine in semantic segmentation branch to extract global level features via RoIAlign. RoIAlign is used to extract all levels of features from feature maps. RPN is used to generate text proposals and forward those to detection and mask branches. For extracting features from feature maps in the detection and mask branches, they perform, what they call, multi-path Fusion, which essentially combines, i.e. sums them element wise, features from different paths into fused features. In the detection branch the authors extract word level features from the proposals and fuse them with global features to perform bounding box regression for words and characters. After detecting characters, they can extract character level features as well and they do this in the mask branch. When they extract all feature levels, they fuse them all together and use them for instance segmentation. In the end, the output of the mask branch is used in combination

Fig. 2: Orderless box discretization: pipeline

with the output of the detection branch to get the final text detection result. This entire pipeline can be seen on Figure 1.

In order to evaluate the model, the authors also require character annotations in the dataset and since most datasets only have word annotations, they generate them on their own. For generating character annotations, their own pre-trained model is used to detect characters which will be used in evaluation, i.e. they perform weak supervision.

Orderless Box Discretization Network

Orderless Box Discrectization Network is a regression-based method for scene text detection [25]. The model outputs quadrilateral labels, since they are much more effective at detecting curved text than rectangular labels. The authors introduce a novel method for regressing the bounding box in which they address the problem of inconsistent labeling. This problem occurs for regression-based methods because the labels for the bounding boxes are highly sensitive to changes in the image, like rotation, so it causes a significant change in label sequence with just small amount of interference. To solve this problem, they introduce a system that predicts order irrelevant points, which they call Key Edges (KE). These KEs are invariant representations of the coordinates which do not affect label sequence and are still able to produce a valid bounding box.

Their system architecture consists of three parts: an Orderless Box Discretization (OBD) block, a matching-type learning (MTL) block and rescoring and post-processing (RPP) block. Input image is first let through FPN and RPN, where the authors generate region of interest proposals from which they extract features with RoIAlign and feed them into the OBD block. The Orderless Box Discretization block is used to generate quadrilateral bounding boxes. Each quadrilateral bounding box is described with eight KEs which make four vertices of the bounding box: minimum x and y, the second-smallest x and y, the second-largest x and y, and the maximum x and y. After the OBD block has generated these eight KEs, they are fed into the matching-type learning block which takes care of the matching between x and y KEs to form the most accurate bounding box. Each x KE should match exactly one y KE and all should be matched where they represent the four vertices of the bounding box. Every slightly different pairing results in a different bounding box and since there is four x KEs trying to be matched with y KEs with respect to their order, there is exactly twenty four possible matchings. To further explain this, first x KE can be matched with one of four y KEs, second x KE can then be matched with remaining three y KEs, third x KE can be matched with the remaining two y KEs and the last x KE can only be matched with the last y KE so in total, when multiplicaion is carried out, there are twenty four combinations. Therefore, the model outputs twenty four classes and the authors train the model by minimizing the cross-entropy loss. Finally, the rescoring and post-processing block is used to reduce the number of false positives from the first two blocks by calculating scores for the KEs and drawing conclusions from them. This pipeline can be seen on Figure 2.

Real-time Scene Text Detection with Differentiable Binarization

This is a segmentation-based method for scene text detection that is characterized by a particularly low inference time and thus high frames per second rate [20]. Segmentation-based methods usually require complex post-processing step and because of that they are slow. These methods produce a probability map which needs to be transformed into binary map by using some fixed threshold and also it is necessary to apply certain pixel-level grouping method to get final detection result. Their proposed method inserts binarization step into the model to achieve simpler post-processing and better inference time.

The idea of involving the binarization into the model is not only lowering the inference time, but also optimizing this post-processing step. By optimizing it inside of a neural network, the authors enable the network to adaptively set the threshold which should perform better than fixed threshold at distinguishing background from the text. Standard process of binarization is not differentiable and so it cannot be learned by the neural networks backpropagation step, so they introduce a function which carries out differentiable binarization. Usually, binarization is performed as follows:

$$B_{i,j} = \begin{cases} 1, & \text{if } P_{i,j} \geq t \\ 0, & \text{otherwise} \end{cases}$$

The new differentiable method is:

$$B_{i,j} = \frac{1}{1 + e^{-k(P_{i,j} - T_{i,j})}}$$

B represents output from the binarization or the binarized map, P is the probability map produced from the segmentation-based method, t is a fixed threshold, k is an amplifying factor and T is the adaptive threshold map.

The architecture of their proposed method is a pipeline consisting of the following steps: feature extraction, generating probability map, generating threshold map and generating approximate binary map. In the feature extraction step, the input image is fed into a FPN backbone where the authors produce different levels of feature maps which

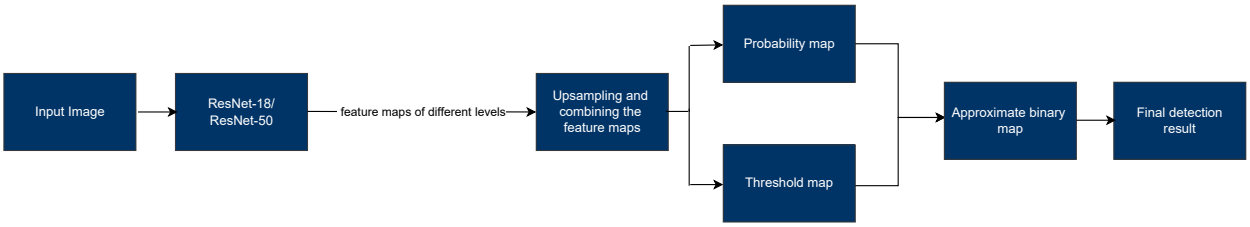

Fig. 3: Segementation-based approach with Differentiable Binarization: pipeline

are then up-sampled to the same scale and combined into a final features used for segmentation. For the backbone, they experimented with ResNet-18 and ResNet-50. Using these features, they generate probability map and threshold map in parallel branches. These maps are, then, used to generate the approximate binary map. Approximate binary map should correctly segment the bounding boxes found in the image. Since they perform supervised training, they generate labels for the probability and threshold maps, but for the binary map the same labels are used as for the probability map.

Progressive Scale Expansion Network

Progressive Scale Expansion Network (PSENet) is a scene text detection model that addresses two issues: detecting text of arbitrary shape and distinguishing text instance from another text instance that is close to it in the image [19]. For the first problem, the authors opt for the pixel-wise segmentation because they believe it performs better than regressing the vertex coordinates of the bounding box. However, segmentation-based methods fail to solve the second problem of separating two really close text instances because pixel-wise segmentation will most likely combine the two text instances into one. PSENet solves this problem using progressive scale expansion algorithm.

PSENet pipeline consists of only two steps: feature extraction and progressive scale expansion block. In the feature extraction step, the input image is fed into a FPN which outputs four levels of feature maps. Low level feature maps are concatenated with high level feature maps and thereby generating four larger feature maps which the authors combine into single feature representation. This feature representation is used for extracting segmentation maps in the progressive scale expansion block. They implement progressive scale expansion block as follows: feature representation extracted from the input image is sent to n number of branches, where each branch is responsible for generating a segmentation map. Each segmentation map masks each text instance in the image, but with different scales. So, for each text instance in the image, they predict multiple segmentation maps, which they call kernels. Each kernel segments the same shape of the input text instance and have the same central point, but the segmented pixels differ in size, i.e. each kernel denotes different area size of pixels for the same instance. Area size of the pixels or the scale of the kernel is a hyperparameter they set orderly where the first kernel returns the smallest scales for the text instances and the last kernel returns the biggest scales of the text instances. After obtaining the kernels, they carry out the core part of the algorithm which is responsible for creating precise segmentation masks for each text instance in the image. This step is based on the Breadth-First-Search (BFS) algorithm, i.e. they perform exhaustive search of the best fit segmentation masks. Here, they start with the first, smallest, kernel where they detect number of text instances or connected components. By doing this, they define the central part of every text instance. After that, they progressively expand other kernels by grouping their pixels and completing the shapes of connected components. Finally they extract these connected components as independent text instances.

To train this model, the authors require the ground truth for the kernels of all selected scales. They generate different scales of the text instance from the bounding box labels by shrinking the bounding box with the Vatti clipping algorithm, but the original bounding box is used as the ground truth for the biggest scale.

IV. Evaluation of the State-of-the-art solutions

In this section, the evaluation of the approaches described in the previous section is carried out. Their performances are compared on ICDAR 2015 benchmark dataset which is described in more detail in the following subsection. Metrics that were used for the evaluation are: precision, recall and F-measure. Also, FPS was compared between the models. The evaluation itself was carried out in the subsection B.

A. Dataset

The ICDAR 2015 dataset, or the International Conference on Document Analysis and Recognition 2015 dataset, is a dataset commonly used for evaluating text detection and recognition methods. It consists of 1500 images, where 1000 images were used for training and 500 for testing [26]. It includes both scene text images and born-digital images. The text in images comes in different languages and orientations and also some of the images intentionally have blur, noise or low resolution. This dataset was used as a benchmark dataset for evaluation of all architectures that are described in this paper.

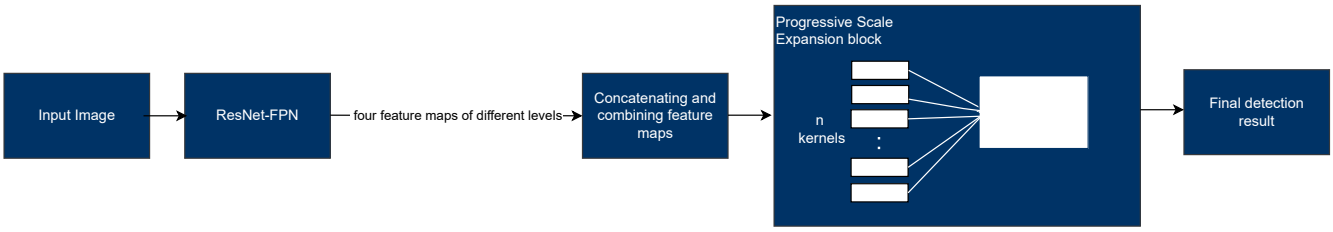

Fig. 4: PSENet: pipeline

Model	Precision	Recall	F-measure
TextFuseNet	**93.96**	**90.56**	**92.233**
OBD	92.1	88.2	90.1
DB	91.8	83.2	87.3
PSENet	88.7	85.5	87.1

TABLE I: Detection results on the ICDAR 2015 dataset

Model	FPS
TextFuseNet	8.3
OBD	4.5
DB	**82**
PSENet	12.38

TABLE II: Highest FPS achieved

B. Comparisons of state-of-the-art approaches on benchmark dataset

In this section, the results of the evaluation are presented. Like mentioned, the approaches are evaluated on the standard metrics: precision, recall and F-measure. The results can be seen in the table 1. TextFuseNet achieved the best measures on all three metrics and therefore perform the best on ICDAR 2015.

Aside from their metrics performance, it is also important that these methods return the localization results quickly, since scene text detection in an outdoor environment is often carried out in real-time and therefore requiring high frames per second (FPS) rate. More complex methods usually achieve better metrics, but worse FPS and simpler methods achieve worse metrics and better FPS. Sometimes, getting better FPS is more important than accuracy and in that case simpler backbones are used for feature extraction. Described approaches experimented with different architectures and have evaluated the models on different datasets than ICDAR 2015. In table 2, one could see the highest FPS rate achieved in each paper, not necessarily on ICDAR 2015. DB model recorded the highest FPS, which was also the selling point of their paper.

V. Conclusion

In this paper, an overview of state-of-the-art solutions for scene text detection was given. These solutions use complex neural networks to solve this task and they can be roughly classified into three groups: regression-based, segmentation-based and hybrid methods. They were evaluated on ICDAR 2015 benchmark dataset with respect to the precision, recall and F-measure. Also, the comparison of the real-time inference was conducted by comparing their FPS rates.

References

[1] "Optical character recognition," 2022, [Online; accessed 14-January-2023]. [Online]. Available: https://en.wikipedia.org/wiki/Optical_character_recognition

[2] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," 2018. [Online]. Available: https://arxiv.org/abs/1811.04256

[3] M. Ibrayim, Y. Li, and A. Hamdulla, "Scene text detection based on two-branch feature extraction," *Sensors*, vol. 22, no. 16, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/16/6262

[4] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," 2016. [Online]. Available: https://arxiv.org/abs/1609.03605

[5] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," 2017. [Online]. Available: https://arxiv.org/abs/1703.06520

[6] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," 2017. [Online]. Available: https://arxiv.org/abs/1709.00138

[7] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," 2016. [Online]. Available: https://arxiv.org/abs/1611.06779

[8] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," 2017. [Online]. Available: https://arxiv.org/abs/1703.08289

[9] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2cnn: Rotational region cnn for orientation robust scene text detection," 2017. [Online]. Available: https://arxiv.org/abs/1706.09579

[10] M. Liao, B. Shi, and X. Bai, "TextBoxes: A single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3676–3690, aug 2018. [Online]. Available: https://doi.org/10.1109%2Ftip.2018.2825107

[11] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," 2018. [Online]. Available: https://arxiv.org/abs/1803.05265

[12] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," 2016. [Online]. Available: https://arxiv.org/abs/1604.04018

[13] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction," 2016. [Online]. Available: https://arxiv.org/abs/1606.09002

[14] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "Wordsup: Exploiting word annotations for character based text detection," 2017. [Online]. Available: https://arxiv.org/abs/1708.06720

[15] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," 2018. [Online]. Available: https://arxiv.org/abs/1807.01544

[16] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," 2018. [Online]. Available: https://arxiv.org/abs/1802.08948

[17] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," 2018. [Online]. Available: https://arxiv.org/abs/1801.01315

[18] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," 2019. [Online]. Available: https://arxiv.org/abs/1908.05900

[19] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," 2019. [Online]. Available: https://arxiv.org/abs/1903.12473

[20] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," 2019. [Online]. Available: https://arxiv.org/abs/1911.08947

[21] J. Liu, X. Liu, J. Sheng, D. Liang, X. Li, and Q. Liu, "Pyramid mask text detector," *CoRR*, vol. abs/1903.11800, 2019. [Online]. Available: http://arxiv.org/abs/1903.11800

[22] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "TextField: Learning a deep direction field for irregular scene text detection," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5566–5579, nov 2019. [Online]. Available: https://doi.org/10.1109%2Ftip.2019.2900589

[23] Y. Liu, S. Zhang, L. Jin, L. Xie, Y. Wu, and Z. Wang, "Omnidirectional scene text detection with sequential-free box discretization," *CoRR*, vol. abs/1906.02371, 2019. [Online]. Available: http://arxiv.org/abs/1906.02371

[24] J. Ye, Z. Chen, J. Liu, and B. Du, "Textfusenet: Scene text detection with richer fused features," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 516–522, main track. [Online]. Available: https://doi.org/10.24963/ijcai.2020/72

[25] Y. Liu, T. He, H. Chen, X. Wang, C. Luo, S. Zhang, C. Shen, and L. Jin, "Exploring the capacity of an orderless box discretization network for multi-orientation scene text detection," 2019. [Online]. Available: https://arxiv.org/abs/1912.09629

[26] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "Icdar 2015 competition on robust reading," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1156–1160.