Learning Robust Feature Representations for Scene Text Detection

Sihwan Kim
BigData and AI Laboratory
Hana Institute of Technology, Hana TI
Seoul 06133, Republic of Korea
shnasc.kim@hanafn.com

Taejang Park*
BigData and AI Laboratory
Hana Institute of Technology, Hana TI
Seoul 06133, Republic of Korea
djang000@hanafn.com

Abstract

Scene text detection based on deep neural networks have progressed substantially over the past years. However, previous state-of-the-art methods may still fall short when dealing with challenging public benchmarks because the performances of algorithm are determined by the robust features extraction and components in network architecture. To address this issue, we will present a network architecture derived from the loss to maximize conditional log-likelihood by optimizing the lower bound with a proper approximate posterior that has shown impressive performance in several generative models. In addition, by extending the layer of latent variables to multiple layers, the network is able to learn robust features on scale with no task-specific regularization or data augmentation. We provide a detailed analysis and show the results on three public benchmark datasets to confirm the efficiency and reliability of the proposed algorithm. In experiments, the proposed algorithm significantly outperforms state-of-the-art methods in terms of both recall and precision. Specifically, it achieves an H-mean of 95.12 and 96.78 on ICDAR 2011 and ICDAR 2013, respectively.

1 Introduction

Scene text detection is one of the most prolific research areas from the computer vision community due to its wide range of applications in multilingual translation, document analysis, scene understanding, autonomous driving, etc. Although previous works have made remarkable progresses, it is still challenging problem because of its various text shapes and highly complicated backgrounds. Recently, the text detectors based on deep learning [1-13] beyond the traditional approaches [14-19] have shown promising performance. While the traditional approaches have been designed to manually extract features of the scene texts, the deep learning based text detectors are designed to learn the features that are deemed useful from the training data. However, instead of designing, the network architecture, most of the text detectors based on convolutional neural network have commonly focused their attention into generating the ground truth and post-processing in order to obtain single text instances.

To address this issue, we will introduce a novel network architecture derived from the loss to maximize conditional log-likelihood for text detection, i.e. \( \log p(y|x) \) where \( y \) and \( x \) are the ground truth and input images, respectively. It can be conducted by optimizing the lower bound with a proper approximate posterior. While the most successful generative models often use only a single layer of latent variables [20], our proposed model uses multiple layers of latent variables to effectively capture meaningful representations at each layer similar to [21,22]. In this process, our model can collect rich features in different resolutions which allow the model to have robustness for scales on the texts.

*Corresponding author

Preprint. Under review.
and objects in the backgrounds. Moreover, we are inspired by the previously developed multi-scale feature fusion techniques \[23–26\] that can effectively aggregate features generated at each layer of the backbone network. Among them, we adopt a weighted bi-directional feature pyramid network \[23\] named BiFPN which shows impressive performance in object detection tasks. Compared to existing methods, the proposed algorithm achieves significantly enhanced performance according to qualitative and quantitative experiments on the public benchmarks. Specifically, our method achieves an H-mean of 95.12 on ICDAR 2011 \[27\] and 96.78 on ICDAR 2013 \[28\], outperforming the previous state-of-the-art methods in terms of both recall and precision.

The contributions of this work are summarized as follow:

- We adopt a multi-scale feature fusion module to extract robust representations on text instances.
- Our method proposes a novel loss scheme for detecting text instances in natural scene images.
- Our method significantly outperforms the previous state-of-the-art method in both recall and precision on ICDAR 2011 and ICDAR 2013 datasets.

2 Related Work

During the past few years, detecting text in the natural scene image has been widely studied in the computer vision area. Before the emergence of deep learning era, most of the text detectors are based on Connected Components Analysis \[14–16\] or Sliding Window \[17–19\]. After the appearance of deep learning based methods, most of the text detectors have adopted deep learning and gradually become the mainstream. The text detectors which are to localize texts in images can be classified into regression-based or segmentation-based approaches.

**Regression-based text detectors:** Inspired by the popular object detection frameworks, such as SSD \[29\] and Faster R-CNN \[30\], which predict the candidates in the form of coordinates of bounding boxes, various text detectors have been proposed. TextBoxes \[12\] that have quite a similar architecture to SSD \[29\] modified convolutional filters and anchor boxes to effectively handle the highly complicated text instances. Rotation Region Proposal Networks (RRPN) \[31\] inspired by Faster R-CNN \[30\] introduced a novel framework to fit oriented text shapes by proposing rotational RPN and RoI pooling.

**Segmentation-based text detectors:** Another popular approach is to regard text detection as semantic segmentation problem. These methods generally give pixel-level output maps indicating the probability of text regions. The previous text detectors \[1, 9, 11, 32, 33\] corresponding to this approach mainly adopt FCN \[34\] and modify it to predict candidate of text region at pixel-level. While the segmentation-based approaches have the advantage of being able to predict text regions directly, identifying the positive regions from the output map into single text instances is not trivial. For example, multiple text instances lying very close to each other may make it hard to separate each instance. To efficiently separate these text instances, most segmentation-based text detectors need to additionally predict for desired single text instances.

The methods described as above have carefully designed the loss functions in terms of the output characteristics and the post-processing methods to obtain single text instances. Meanwhile, we derive the loss function in a new direction deviated from the conventional perspective as the previous methods. We focus on how to design the network architecture to reflect the loss function which is derived to maximize conditional log-likelihood. Our approach can be applied to all segmentation-based networks with slight architecture modifications because it is independent of the output characteristic and post-processing. To verify this, we will apply this idea to the network architecture of CRAFT \[1\], which achieved state-of-the-art from the ICDAR 2013 dataset, and demonstrate the reliability and performance of our method with three public benchmarks, including ICDAR 2013.

3 Proposed method

In this section, we describe the framework of our proposed model and analyze the loss function in detail. Our network is a fully-convolutional neural network that outputs dense, per-pixel score maps
named region score to localize each individual character and affinity score to link the semantically nearby characters as proposed in CRAFT [1].

3.1 The lower bound of conditional log-likelihood

Let \( \mathbf{X} = \{ (\mathbf{x}_j, y_j)_j \}_{j=1}^N \) be the training image datasets, where \( \mathbf{x}_j \) is the input image and \( y_j \) represents both the region and affinity score maps, consisting of \( N \) i.i.d. observations. We introduce latent variables \( \mathbf{z} = \{ z_1, \ldots, z_L \} \) where \( z_i \) represents the layer closest to \( \mathbf{x} \) and \( z_L \) represents the top layer. Extending a single layer of latent variables for abstract feature representations allows the model to utilize deeper and more distributed latent variables [21] [22]. Hence, we carefully design a network architecture to learn features of different levels between the dataset \( \mathbf{X} \) and the latent variables \( \mathbf{z} \). In our case, we assume that the latent variables \( \mathbf{z} \) are independent for each variable \( z_i \). We wish to maximize the conditional log-likelihood \( \log p(y|x) \) over a datasets \( \mathbf{X} \). Formally, we would like to maximize

\[
\log p(y|x) = \sum_{j=1}^N \log p(y_j|x_j),
\]

which is non-convex and often intractable in general. To avoid the intractable integral, we introduce an approximate posterior \( q(z|x_j) \) to obtain the lower bound using Jensen’s inequality and the fact that the integral of probability distribution equal to 1.

\[
\log p(y_j|x_j) \geq \int \log \left( p(y_j|z, x_j) \frac{p(z|x_j)}{q(z|x_j)} \right) q(z|x_j) \, dz,
\]

\[
= \int \log \left( p(y_j|z, x_j) \right) q(z|x_j) \, dz - \int \log \left( \frac{\prod_{l=1}^L q(z_l|x_j)}{p(z_l|x_j)} \right) q(z|x_j) \, dz,
\]

\[
= \mathbb{E}_{q(z|x_j)} \left[ \log p(y_j|z, x_j) \right] - \sum_{l=1}^L D_{KL} \left( q(z_l|x_j) \parallel p(z_l|x_j) \right),
\]

where \( D_{KL} \) denotes Kullback-Leibler divergence. The next sub-section describes the network architecture and loss function in detail.

3.1.1 Network Architecture

We adopt a fully convolutional network architecture based on VGG-16 [35] with batch normalization except for the last three fully-connected layers as our backbone. In addition, we adopt the BiFPN modules that efficiently aggregates the multi-scale features with different resolutions to make a list of new features. Formally, given features \( f^{in} = (f_1^{in}, \ldots, f_5^{in}) \) where \( f_l^{in} \) represents the feature generated from level \( l \) with resolution of \( 1/2^l \) of the input images. We would like to make new features \( f^{out} = (f_1^{out}, \ldots, f_5^{out}) \) by a transformation \( F \) that can effectively aggregate multi-scale features. A common technique when fusing multiple input features with different resolutions is to resize them to the same resolution and then sum all up with equal weights [36] [37]. However, we intuitively believe that the input features with different resolutions could contribute to the output unequally. To deal with this issue, the weighted feature fusion techniques have been developed in [23]. Among them, we will adopt Fast Normalized Fusion technique that the weight \( w_l \geq 0 \) are ensured by applying a ReLU to reduce computational cost while keeping the benefits of normalized weights. For a concrete example, we describe the case of level 4 which is in blue circles in the Figure [1].

\[
f_{4}^{id} = \text{Conv} \left( \frac{w_1 f_4^{in} + w_2 \text{Resize}(f_5^{in})}{w_1 + w_2 + \epsilon} \right),
\]

\[
f_{4}^{out} = \text{Conv} \left( \frac{w_1 f_4^{in} + w_2 f_4^{ld} + w_3 \text{Resize}(f_3^{out})}{w_1 + w_2 + w_3 + \epsilon} \right),
\]

where \( \epsilon = 0.0001 \) is a small value to avoid numerical instability, Resize is conventional upsampling or downsampling operation, \( f_{4}^{id} \) and \( f_{4}^{out} \) are the intermediate and output feature at level 4, respectively. Conv implies Convolution-Batch normalization-ReLU operation consisting of \( 1 \times 1 \) and \( 3 \times 3 \) receptive fields with 128 and 64 filters, respectively.
The generated new features feed into channel bottlenecks similar to a standard auto-encoder that only transforms the dimension of a channel to extract latent variables \( z_j, j = 1, \ldots, 5 \). The reduction part have 3 layers with 32, 16 and 2 filters consisting of \( 1 \times 1 \) and \( 3 \times 3 \) receptive fields each. The expansion part consists of transformed structure of the reduction part. The latter part of our network refines the outputs from the channel bottlenecks into two score maps: the region score and affinity score.

### 3.1.2 Loss Function

We analyze the loss function described in (1) in detail. The most critical point to be determined is how to assign a proper probability distribution to the latent space of the input data. Multivariate Gaussian distribution to real valued attributes or independent Bernoulli distribution to each binary attributes have been generally used in such cases.

Let \( F_{id} : \mathbb{R}^D \rightarrow \mathbb{R}^D \) be an identity function such that \( F_{id}(x_j) = y_j \). Also denote \( F_{ip}^S : \mathbb{R}^T \rightarrow \mathbb{R}^S \) be an interpolation function by resampling using pixel area relations to prevent aliasing phenomenon when a image is downsampled. In our case, the (conditional) prior \( p(z_i | x_j) \) and the model \( q(z_i | x_j) \) can both be modeled using Bernoulli distributions:

\[
p(z_i | x_j) = \prod_{k=1}^{Q(l)} \text{Ber}(z_{ij}^k; F_{ip}^{Q(l)} \circ F_{id}(x_j)), \quad q(z_i | x_j) = \prod_{k=1}^{Q(l)} \text{Ber}(z_{ij}^k; \theta^k(x_j)),
\]

where \( Q(l) \) is the dimension of the latent variables at \( l \)-th layer and \( \theta^k(x_j) \) are predicted by the model \( q(z_i | x_j) \). For convenience, we use notation \( y_{ij,l} \) to denote the mapping \( F_{ip}^{Q(l)} \circ F_{id}(x_j) \). The first term of (1) can be calculated by the Stochastic Gradient Variational Bayes (SGVB) where the Monte Carlo estimates are performed with the reparameterization trick. For the Bernoulli distribution case, the reparameterization trick can be performed using a uniformly distributed auxiliary variable \( \epsilon \sim U(0, 1) \) described in [38-39]. A single sample is often sufficient to form the Monte Carlo estimates in practice, thus, we get

\[
\mathbb{E}_{q(z_i | x_j)} \left[ \log p(y_j | z_i, x_j) \right] = \sum_{k=1}^{D} y_{j}^{k} \log p_{j}^{k} + \left( 1 - y_{j}^{k} \right) \log (1 - p_{j}^{k}),
\]
where $y_{k}^{l}$ and $p_{k}^{l}$ are $k$-th element of the ground truths and the estimated outputs, respectively. KL divergence in the second term of (1) is analytically computed as follow.

$$- \sum_{l=1}^{L} D_{KL}(q(z_{l}|x_{j}) \parallel p(z_{l}|x_{j})) = \sum_{l=1}^{L} \int q(z_{l}|x_{j})(\log p(z_{l}|x_{j}) - \log q(z_{l}|x_{j})) \, dz_{l},$$

$$= \sum_{l=1}^{L} E_{q(z_{l}|x_{j})}[\log p(z_{l}|x_{j})] - E_{q(z_{l}|x_{j})}[\log q(z_{l}|x_{j})],$$

$$= \sum_{l=1}^{L} \sum_{k=1}^{Q(l)} z_{k}^{l} \log \frac{y_{k,j}^{l}}{z_{l}^{k}} + (1 - z_{k}^{l}) \log \frac{1 - y_{k,j}^{l}}{1 - z_{l}^{k}},$$

(3)

where $z_{k}^{l}$ and $y_{k,j}^{l}$ are $k$-th element of the latent variables and the transformed value of $x_{j}$ by the mapping defined as above at $l$-th layer, respectively. By taking the sum of (2) and (3), we get

$$\log p(y_{j}|x_{j}) \geq L_{\text{recons}} + L_{\text{reg}},$$

(4)

where $L_{\text{recons}}$ and $L_{\text{reg}}$ are reconstruction loss (2) and regularization loss (3), respectively.

Our proposed model will be trained to minimize the negative log-likelihood defined by (4), which implies minimizing the reconstruction loss (2) and the regularization loss (3). The regularization loss encourages the approximate posterior $q(z_{l}|x_{j})$ to be close to the Bernoulli distribution $p(z_{l}|x_{j})$ defined by the resized ground truth with the shape corresponding to each $l$-th layer. Thus, due to the regularization loss, when the network is fully trained, the defined distribution at each $l$-th layer is naturally induced in our model. This implies that our model is able to learn robust representations for scales on texts and objects in backgrounds.

### 3.2 Label Generation

For generating region and affinity score maps, which represent the probability of the center of the character and the relationships between semantically nearby characters in single text instances, respectively, we adopt the method of label generation described in CRAFT [1]. Although ICDAR 2011 and ICDAR 2013 datasets do not provide character-level annotations directly, it is possible to generate ground truth labels by using extra information provided for text segmentation. In the case of ICDAR 2011, the dataset for text segmentation is provided in the form of images where white pixels are interpreted as the background while non-white pixels are color coded so that pixels of each character in the image are shown in the same color. Hence the individual characters in word-level bounding box can be easily separated by the color identity. For ICDAR 2013, which is analogous to ICDAR 2011, the ground truth label can be also generated by separating characters in the images with the dataset provided for text segmentation. The examples of generated ground truth labels for the case of ICDAR 2011 and ICDAR 2013 are illustrated in Figure 2.
4 Experiments

To verify the reliability of our proposed method, we used four public benchmarks, which are briefly introduced below, three of which are used to evaluate our method.

4.1 Benchmark Datasets

**SynthText** [40] is a large scale datasets that contains about 800K synthetic images. These images are synthetically created by blending rendered words with natural images. This dataset is used for pre-training our model.

**ICDAR 2011** [27] is proposed in the Challenge 1 of the 2011 Robust Reading Competition for Web and Email scene text detection, consisting of 410 training images and 141 testing images. The text regions are annotated in the form of axis-aligned bounding boxes.

**ICDAR 2013** [28] is a datasets proposed in the Challenge 2 of the 2013 Robust Reading Competition for focused horizontal text detection in scenes. It consists of 229 images and 233 images for training and testing, respectively. The text regions are annotated in the form of axis-aligned bounding boxes.

**ICDAR 2015** [41] is used in Challenge 4 of the 2015 Robust Reading Competition. It includes a total of 1500 scene images consisting of 1000 training images and 500 testing images with annotations labeled as word level quadrangles.

4.2 Implementation Details

The network is pre-trained on SynthText for 50K iterations and fine-tuned it on both ICDAR 2011 and ICDAR 2013 together. The longer side of the training images are resized to 640, preserving their aspect ratio. We adopt Adam optimizer [42] as our learning rate scheme. The learning rate is set to $10^{-3}$ initially and decays exponentially with a rate of 0.95 every 1K iterations on both pre-training and fine-tuning stage. We train our model with the batch size of 32 on 8 GPUs in parallel, i.e. 4 images per GPU, and use synchronized batch normalization to effectively calculate the moving average and moving variance. To resolve the imbalance between positive and negative samples when calculating the loss defined by [4], we use the Online Hard Negative Mining proposed in [43] and set the ratio of positives to negatives to 1:3. Also, the data augmentation techniques used to train the model are: (1) random rotation with an angular range of -10 to 10; (2) random flipping with a probability of 0.5; (3) color-channel swapping with a probability of 0.5.

During the testing, the estimated region score and affinity score are binarized with the value $\tau_r$ and $\tau_a$ where $\tau_r$ and $\tau_a$ are the region and affinity threshold, respectively. By taking the boolean addition of these binarized score maps, we get the final output map $M$. Then Connected Component Labeling (CCL) is performed on $M$ and single text instances are obtained by finding each quadrilateral bounding box enclosing the minimum area of each label.

4.3 Experiment Results

We evaluate the proposed method on three challenging public benchmarks: ICDAR 2011 [27], ICDAR 2013 [28], and ICDAR 2015 [41]. We use both ICDAR 2011 and ICDAR 2013 datasets for training, and ICDAR 2015 dataset is only used for testing as unseen data because ICDAR 2015 does not provide character-level annotations. We analyze the results both qualitatively and quantitatively to confirm the efficiency and reliability of our proposed method.

4.3.1 Qualitative Results

Several detection examples on ICDAR 2011, ICDAR 2013 and ICDAR 2015 datasets are shown in Figure 3. As shown, the results demonstrate that our proposed method has strong capability for detecting extremely ambiguous text lines and perspective distortion. However, our method still has a gap to achieve a perfect performance. Several failure cases are illustrated in Figure 3(d). False positives and missing texts may appear in certain situations, such as non-uniform illumination, curved texts and occluded backgrounds. The main contribution of our method is to enable the model to effectively extract robust features related to text instances owing to the multi-scale feature fusion module and the loss function. In order to verify the effectiveness of our method, the extracted feature
maps of each level are displayed in Figure 4. As can be seen in Figure 4, our model captures the local structures of text instances in the lower level layers and features with semantic information are extracted in the higher level layers. Additionally, we illustrate the learned normalized weights for $f_{\text{out}}^l$ at three different BiFPN modules in Figure 5 to further understand the behavior of Fast Normalized Fusion. As we expected, the normalized weights change during training which implies there exists suitable weights for the different features representing their contribution to the output.

4.3.2 Quantitative Results

We compared our proposed model with the previous state-of-the-art methods on three public benchmarks (ICDAR 2011, ICDAR 2013 and ICDAR 2015). All experiments are performed with single image resolution that the longer side of the images in ICDAR 2011 and ICDAR 2013 are resized to 640 and that of ICDAR 2015 is resized to 1024. The image size are set differently since ICDAR 2015 has many small-scale texts. The performance of our method as well as other methods on three datasets are presented in Table 1. Our method achieves state-of-the-art performance on ICDAR 2011 and ICDAR 2013 dataset, which demonstrates our approach extracts more robust features on text instances compared to other methods. In the case of ICDAR 2015, since our model trained only on ICDAR 2011 and ICDAR 2013 dataset due to the lack of character-level annotation in ICDAR 2015, the performance appear to be lower than other previous methods. However, when compared with the method such as Zhang et al. [36], CTPN [44] and Yao et al. [32], our method is still competitive and leads the model to learn generalized feature representations for text detection.

Furthermore, to highlight the effectiveness of the BiFPN module and the derived loss (4), we finally perform an ablation study to compare the performances of our method with the following three models: (1) Baseline: The model excluding both the BiFPN modules and the channel bottlenecks equipped with the regularization loss, (2) Baseline + BiFPN: The model excluding only the channel
Figure 5: The illustration of learned normalized weights for \( f_{\text{out}} \) at three different BiFPN modules. The weights for input1 and input2 are represented in red and blue, respectively.

Table 1: Results on three public benchmarks: ICDAR 2011, ICDAR 2013 and ICDAR 2015. \( R \), \( P \) and \( H \) represent recall, precision and H-mean, respectively. The best score is highlighted in bold. The asterisk(\( * \)) denotes the results from the unseen dataset.

| Method                  | ICDAR 2011 (DetEval) | ICDAR 2013 (DetEval) | ICDAR 2015 (DetEval) |
|-------------------------|-----------------------|-----------------------|-----------------------|
|                         | R        | P        | H        | R        | P        | H        | R        | P        | H        |
| Pal_DAS16 [45]          | 87.95    | 91.14    | 89.51    | -        | -        | -        | -        | -        | -        |
| SCUT_HCII [46]          | 81       | 85       | 83       | 72.89    | 92.79    | 81.65    | -        | -        | -        |
| Text-CNN [47]           | 74       | 91       | 82       | 80.80    | 91.31    | 85.73    | 80.55    | 86.59    | 83.46    |
| Ruan et al. [48]        | 90.71    | 95.43    | 93.01    | -        | -        | -        | -        | -        | -        |
| Zhang et al. [36]       | 78       | 88       | 83       | 78       | 88       | 83       | 43       | 71       | 54       |
| CTPN [44]               | 83       | 93       | 88       | 83       | 93       | 88       | 52       | 74       | 61       |
| Yao et al. [32]         | 80.2     | 88.8     | 84.3     | 80.2     | 88.8     | 84.3     | 58.7     | 72.3     | 64.8     |
| SegLink [6]             | 83       | 87.7     | 85.3     | 83       | 87.7     | 85.3     | 76.8     | 73.1     | 75       |
| Texboxes++ [13]         | 86       | 92       | 89       | 86       | 92       | 89       | 78.5     | 87.8     | 82.9     |
| RRPN [31]               | 57.31    | 95.19    | 91.08    | 57.31    | 95.19    | 91.08    | 77.13    | 83.52    | 80.20    |
| Stela [10]              | 85.1     | 93.3     | 89       | 85.1     | 93.3     | 89       | 78.58    | 88.7     | 83.3     |
| Mask Textspotter [7]    | 88.1     | 94.1     | 91       | 88.1     | 94.1     | 91       | 81.2     | 85.8     | 83.4     |
| FOTS [5]                | 90.47    | 94.63    | 92.5     | 90.47    | 94.63    | 92.5     | 85.17    | 91       | 87.99    |
| CRAFT [11]              | 93.06    | 97.43    | 95.20    | 93.06    | 97.43    | 95.20    | 84.26    | 89.79    | 86.93    |
| Baseline                | 92.68    | 94.24    | 93.45    | 93.15    | 97.50    | 95.27    | 53.64*   | 73.00*   | 61.84*   |
| Baseline + BiFPN        | 93.52    | 95.67    | 94.58    | 94.28    | 98.02    | 96.12    | 60.13*   | 68.36*   | 63.99*   |
| Baseline + KL           | 94.23    | 95.04    | 94.64    | 94.39    | 98.34    | 96.32    | 57.39*   | 71.98*   | 63.86*   |
| Proposed                | 94.35    | 95.90    | 95.12    | 95.14    | 98.48    | 96.78    | 61.24*   | 78.33*   | 68.74*   |

bottlenecks, (3) Baseline + KL : The model excluding only the BiFPN modules. As shown in Table 1, the performance results of all four models, including our proposed model, improve gradually from the baseline to our model in both recall and precision. The remarkable performance improvement between the baseline and our proposed model, especially in ICDAR 2015, implies that our method encourages the enrichment of feature representations on text instances.

5 Conclusion and Future Work

In this paper, we systematically designed a network architecture to maximize conditional log-likelihood and adopted a multi-scale feature fusion module to effectively aggregate the features generated from the backbone network. The proposed method achieves state-of-the-art performances on two public benchmarks (ICDAR 2011 and ICDAR 2013) and shows the generalization ability through ICDAR 2015 considering the unseen data. We believe that our proposed network architecture can be applied to all segmentation-based model and can lead to additional performance improvements. For the potential future work, we will apply the proposed network architecture to other previous models based on word-level annotation and continue to develop a better design of the networks.
References

[1] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee. Character region awareness for text detection. *In CVPR*, pages 9365–9374, 2019.

[2] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li. Single shot text detetor with regional attention. *In ICCV*, 6, 2017.

[3] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo. R2cnn: rotational cnn for orientation robust scene text detection. *arXiv preprint:1706.09579*, 2017.

[4] X. Zhou, C. Yao, H. Wen, Y. Wang, and S. Zhou. East: an efficient and accurate scene text detector. *In CVPR*, pages 5551–5560, 2017.

[5] X. Liu, D. Liang, S. Yan, D. Chen, and Y. Qiao. Fots: Fast oriented text spotting with a unified network. *In CVPR*, pages 5676–5685, 2018.

[6] B. Shi, X. Bai, and S. Belongie. Detecting oriented text in natural images by linking segments. *In CVPR*, pages 3482–3490, 2017.

[7] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *arXiv preprint:1706.09579*, 2017.

[8] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai. Multi-oriented scene text detection via corner localization and region segmentation. *In CVPR*, pages 7553–7563, 2018.

[9] D. Deng, H. Liu, X. Li, and D. Cai. Pixellink: detecting scene text via instance segmentation. *In AAAI*, 2018.

[10] L. Deng, Y. Gong, X. Lu, Y. Lin, Z. Ma, and M. Xie. Stela: A real-time scene text detector with learned anchor. *IEEE Access*, 7:153400–153407, 2019.

[11] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. *arXiv preprint:1807.01544*, 2018.

[12] M. Liao, B. Shi, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. *In AAAI*, 2017.

[13] M. Liao, B. Shi, and X. Bai. Textboxes++: A single-shot oriented scene text detector. *Image Processing*, 27(8):3676–3690, 2018.

[14] M. Bušta, L. Neumann, and J. Matas. Fastext: Efficient unconstrained scene text detector. *In ICCV*, 2015.

[15] L. Neumann and J. Matas. Scene text localization and recognition with oriented stroke detection. *In ICCV*, pages 97–104, 2013.

[16] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. *In ICCV*, 2015.

[17] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. *In CVPR*, pages 366–373, 2004.

[18] M Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. *In ECCV*, pages 512–528, 2014.

[19] S. Zhu and R. Zanibbi. A text detection system for natural scenes with convolutional feature learning and cascaded classification. *In CVPR*, pages 625–632, 2016.

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint:1312.6114*, 2013.

[21] S. Zhao, J. Song, and S. Ermon. Learning hierarchical features from deep generative models. *In ICML*, 70:4091–4099, 2017.

[22] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. *In NIPS*, pages 3738–3746, 2016.

[23] M. Tan, R. Pang, and Q. V. Le. Efficientdet: scalable and efficient object detection. *arXiv preprint:1911.09070*, 2019.

[24] T. Y. Lin, P. Dollár, R. Girshick, K. He, and B. Hariharan. Feature pyramid networks for object detection. *In CVPR*, pages 2117–2125, 2017.
[25] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. *In CVPR*, pages 8759–8768, 2018.

[26] G. Ghiasi, T. Y. Lin, and Q. V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. *In CVPR*, pages 7036–7045, 2019.

[27] D. Karatzas, S. Robles Mestre, J. Mas, F. Nourbakhsh, and P. Pratim Roy. Icdar 2011 robust reading competition challenge 1: reading text in born-digital images (web and email). *2011 International Conference on Document Analysis and Recognition*, pages 1485–1490, 2011.

[28] D. Karatzas, F. Shafait, U. Hchida, 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 ICDAR, IEEE*, pages 1484–1493, 2013.

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. *In ECCV*, pages 21–37, 2016.

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *PAMI*, 6:1137–1149, 2017.

[31] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transaction on Multimedia*, 20:3111–3122, 2018.

[32] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. Scene text detection via holistic, multi-channel prediction. *arXiv preprint:1606.09002*, 2016.

[33] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao. Shape robust text detection with progressive scale expansion network. *arXiv preprint:1903.12473*, 2019.

[34] J. Long, E. Shelhamer, and T. Darrel. Fully convolutional networks for semantic segmentation. *In CVPR*, pages 3431–3440, 2015.

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *In ICLR*, 2015.

[36] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu. Scrdet: Towards more robust detection for small, cluttered and rotated objects. *In ICCV*, pages 8232–8241, 2019.

[37] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint:1611.01144*, 2016.

[38] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. *In CVPR*, pages 2315–2324, 2016.

[39] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. Chandrasekhar, S. Lu, and et al. icdar 2015 competition on robust reading. *In ICDAR, IEEE*, pages 1156–1160, 2015.

[40] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint:1412.6980*, 2014.

[41] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. *In CVPR*, pages 3482–3490, 2017.

[42] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network. *In ECCV*, pages 56–72, 2016.

[43] K. Chen, F. Yin, and C. L. Liu. Effective candidate component extraction for text localization in born-digital images by combining text contours and stroke interior regions. *Document Analysis Systems*, pages 352–357, 2016.

[44] Z. Y. Zhong, L. W. Jin, S. Y. Zhang, and Z. Y. Feng. Deeptext: A unified framework for text proposal generation and text detection in natural images. *arXiv preprint:1605.07314*, 2016.

[45] T. He, W. L. Huang, Y. Qiao, and J. Yao. Text-attentional convolutional neural network for scene text detection. *IEEE transactions on image processing*, 25(6):2529–2541, 2016.

[46] S. Ruan, J. Lu, F. Xie, and Z. Jin. A novel method for fast arbitrary-oriented scene text detection. *In 2018 Chinese Control And Decision Conference(CCDC)*, pages 1652–1657, 2018.