A Learning-based Text Synthesis Engine for Scene Text Detection

Xiao Yang[1]
xuy111@psu.edu
Dafang He[2]
duh188@psu.edu
Daniel Kifer[1]
dkifer@cse.psu.edu
C. Lee Giles[2]
giles@ist.psu.edu

[1] Department of Computer Science and Engineering
Pennsylvania State University

[2] College of Information Science and Technology
Pennsylvania State University

Abstract

Scene text detection (STD) and recognition (STR) methods have recently greatly improved with the use of synthetic training data playing an important role. That being said, for text detection task the performance of a model that is trained sorely on large-scale synthetic data is significantly worse than one trained on a few real-world data samples. However, state-of-the-art performance on text recognition can be achieved by only training on synthetic data [10]. This shows the limitations in only using large-scale synthetic data for scene text detection. In this work, we propose the first learning-based, data-driven text synthesis engine for scene text detection task. Our text synthesis engine is decomposed into two modules: 1) a *location* module that learns the distribution of text locations on the image plane, and 2) an *appearance* module that translates the text-inserted images to realistic-looking ones that are essentially indistinguishable from real-world scene text images. Evaluation of our created synthetic data on ICDAR 2015 Incidental Scene Text dataset [15] outperforms previous text synthesis methods.

1 Introduction

Automatic localizing and recognition of text in natural scene images — text spotting — is still an open area of research. It consists of two sub-tasks: 1) finding the text region in the image, or *text detection*, and 2) recognition of the text in a cropped image patch, or *text recognition*. A text spotting system with high performance would have broad applications such as autonomous driving vehicles (responsively to street signs) and assistive technologies such as indoor navigation for the visually impaired.

With the use of deep convolutional neural networks, text spotting systems have seen great improvement in performance. However, deep learning methods are usually data hungry and their performance often positively correlates with the amount of available training data. To tackle this problem, researchers in text spotting community [7, 10, 28, 31] proposed the use of large-scale synthetic data as a training set. This way, large amounts of training data with

Train Set	#Train	Test Set	#Test	Recall	Precision	F1
IC13+15	1,229	IC15	500	**77.76**	**84.35**	**81.18**
SynthText [7]	858,750	IC15	500	50.48	57.51	53.89

Table 1: Performance of a baseline text detector (EAST [33]) when training on real images or synthetic images. IC13 and IC15 are short for ICDAR 2013 and 2015 dataset, respectively.

detailed annotations are readily available making state-of-the-art performance achievable [7, 10, 12, 25, 28, 30].

However, for the text detection sub-task synthetic data are not as helpful as for text recognition sub-task. Jaderberg et al. [10] trained a text recognizer using 9 millions synthetic word images, and outperformed all previous methods up to that time. In contrast, Gupta et al. [7] generated 8 millions synthetic images for text detection, but the performance when training only on the synthetic data is significantly worse than that on real-world data. Table 1 shows the performance of a baseline text detection model (EAST [33]) when trained on 1) real data (the training set of ICDAR 2013 [14] and 2015 [15]) and 2) synthetic data [7]. Considering the fact that the total number of synthetic data samples (858,750) is orders of magnitude larger than that of real samples (1,229), the performance gap is surprisingly large.

This motivates us to find a better way to generate synthetic images for text detection. Prior work on synthetic text generation often consists of two steps: 1) randomly or selectively put text instances on the image plane, and 2) add certain effects to the inserted text instances, such as blurring or perspective distortion. However, all these steps require sophisticated design of heuristic rules, which may not be optimal.

In this work, we proposed a learning-based, data-driven text synthesis engine. Generating realistic text in the wild images can be decomposed into two sub-tasks: 1) determine the location of the text bounding boxes and 2) make the appearance of the inserted text more realistic. For the first task, we propose a *location* module based on variational auto-encoders. Given a set of real images $\{x\}$ and the corresponding text locations $\{A_x\}$ in each image, the location module aims at learning the conditional distribution $p(A_x|x)$. In this way, we are able to sample plausible text locations when providing a new background image. These sampled text locations indicate where text instances can be put on the background image. For the second task, we propose an *appearance* module which extends the Cycle-GAN [34] model. After compositing text instances on the background images, our goal is to learn a mapping from synthetic data domain X to real data domain Y, such that the composition results look indistinguishable to real-world ones. Cycle-GAN is adopted since it does not require paired training samples. As we are able to obtain a mask for each image that highlights the text regions, we propose an invariant loss to regularize the generators of Cycle-GAN. The invariant loss encourages that the non-text region has fewer changes, while the text region can attain more focus during optimization.

Our main contributions are summarized as follows: 1) we propose the first learning-based, data-driven text synthesis engine. It excludes the needs to design complicated rules when generating synthetic text in the wild images. 2) We evaluate our synthesis engine on ICDAR 2015 dataset, and outperforms the text synthesis engine proposed by [7]. Adding our synthetic data to the training set of ICDAR 2013 and ICDAR 2015 can further improve the performance, which demonstrates the usefulness of our synthesis engine.

2 Related Work

Text Detection and Object Detection: There exists a large body of literature which deals with text detection. Most text detection approaches can be categorized into two classes: character detection based approaches and word detection based approaches [32]. Character detection based approaches first detect individual characters, then group characters into words based on heuristic rules or sophisticated clustering/grouping models. Early efforts on this type of approaches focus on designing sophisticated features such as maximally stable extremal regions (MSER) [7] or stroke-width transform (SWT) [4] to detect characters, while recently deep convolutional neural networks are employed as a text/non-text classifier to find characters in a sliding window manner [11, 28]. Although character detection can often achieve high recall, the precision is often low due to the fact that many characters like "I" or "l" can not be well distinguished from stroke-like noises. Therefore, several sequential steps [8, 11] are necessary to filter false positives and group characters. These steps inevitably accumulate errors and slow down the whole pipeline.

The second type of approaches attempts to detect words directly. Jaderberg et al. [12] first detect candidate word bounding boxes using edge boxes [35] detectors. Each candidate box is further filtered by a random forest classifier and refined by a CNN-based regressor. Gupta et al. [7] present a fully-convolutional regression network that jointly predicts text existence and regresses bounding boxes. Many researchers [13, 21, 26] adapt methods for general-purpose object detection to text detection and achieved good performance. For example, Tian et al. [26] combine Faster R-CNN [23] architecture and a recurrent neural network (RNN) to jointly predict text/non-text score and refine bounding boxes. Ma et al. [21] extend Faster R-CNN to detect rotated text by having extra anchor boxes of different orientations and aspects. Liao et al. exploit SSD [20] object detector to obtain horizontal bounding boxes for words. Other examples include [9, 13, 18, 19, 25].

Synthetic Text Generation: Many researchers proposed using synthetic text during training, since they can provide detailed and cheap annotations at large-scale. For example, Wang el al. [28] generated image patches of characters and used them to train a character classifier. Jaderberg et al. [11] used a set of pre-defined rules (e.g. random font selection, foreground-background composition, random perspective projection) to generate images of words for training. However, the synthesis process of all these methods [11, 28, 30] are rule based and no learning is involved. Gupta et al. [7] attempted to better understand the background image before compositing the foreground (text). They first learn the color-texture segmentation and the pixel-wise depth, then render text on the background image following a set of rules such that the rendered images look realistic. Nevertheless, the placement of the text locations is still not optimized via learning. Their method can be seen as a sophisticated rule-based method. On the contrary, we propose directly learning the distribution of text locations given an background image. Experiments show that our method leads to better detection performance than that in [7].

Variational Auto-Encoder (VAE) and Cycle-GAN Kingma et al. [16] proposed variational auto-encoder and used the re-parameterization trick to enable gradient based optimization. Since then, VAE has been widely used as a generative models [17, 22, 27, 29]. Our work is in line with prior works on conditional VAE [29], where the decoder is also conditioned on a side input, such as attribute or label information in [29] and the background image information in our case.

Generative adversarial network (GAN) [5] is another popular type of generative model. Prior work [1, 3, 5, 6] has shown that GAN can generate sharp images due to the use of

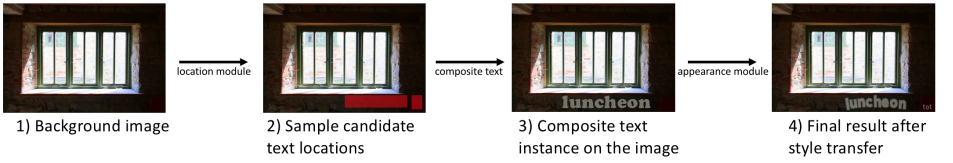

Figure 1: Our text synthesis pipeline. Given a background image, we first sample candidate text locations based on our location module (Section 3.1), then composite text instances according to these locations. The text-inserted images are later fed into our appearance module (Section 3.2) to generate realistic-looking scene text images.

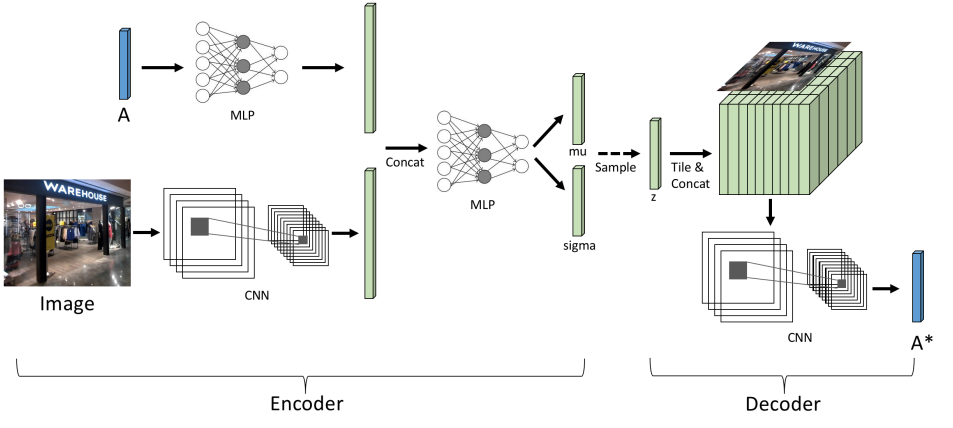

Figure 2: Architecture of the location module. It is a conditional variational auto-encoder that learns the distribution of text locations given background images.

the adversarial loss instead of the L1 loss. Cycle-GAN [34] is a special type of GAN which is designed for image-to-image translation task. It contains two separate generators for the purpose of translating image from domain X to Y ($G : X \rightarrow Y$) and from Y to X ($F : Y \rightarrow X$), and two corresponding discriminators. A cycle-consistency loss is introduced to encourage $F(G(X)) \approx X$ and vice versa. Our work extends this model by introducing an invariant loss. Given a mask M of an image $x \in X$ that highlights the object-of-interest, our invariant loss discourages changes on the background and focus more on the object-of-interest.

3 Method

In this section we describe the details of our learning based text synthesis engine. It consists of two sub modules: a location module and an appearance module. The location module learns the distribution of text locations conditioned on the background image. The appearance module translates the text-inserted images to realistic looking ones. Figure 1 illustrates the whole process. Our text synthesis engine is learning-based, data-driven and excludes the design of complicated heuristic rules. Experiments on ICDAR 2015 Incidental Scene Text dataset [15] demonstrate the superiority of our method compared to prior works.

3.1 The Location Module

Given a background image x, the location module aims to learn the distribution of text locations conditioned on x. In this way, we are able to sample possible text locations that are semantically plausible with respect to the visual context. Each text location is represented by an affine transformation matrix A, which will be used to affine transform a fixed reference box (e.g. a 10-by-10 box in the top-left corner of the image plane). In this way, one affine transformation matrix A uniquely defines one location by affine transforming the reference box.

Inspired by the variational auto-encoder [16, 24], we introduce a latent variable z, and our goal becomes modeling the conditional generative process $p_\theta(A|x,z)$. Note that the notation is slightly different from conventional variational auto-encoders. Here A is the target to be generated instead of x. Here, p_θ is referred to as the decoder, parameterized by θ. During testing, we simply generate text regions in a two-step process:

1. randomly sample latent variable z from a prior distribution (e.g. $\mathcal{N}(0,I)$)

2. take z and a background image x as input, generate text location (represented by an affine transformation matrix A) from $p_\theta(A|x,z)$.

To estimate the parameter θ we would ideally maximize the log likelihood $\log p_\theta(A|x)$. However, it is often intractable to solve such optimization. Kingma et al. [16] introduced another distribution $q_\phi(z|A,x)$ to approximate the true posterior $p_\theta(z|A,x)$, and proposed to maximize the variational lower bound:

$$\log p_\theta(A|x) \geq \mathbb{E}_{q_\phi(z|A,x)}\left[-\log q_\phi(z|A,x) + \log p_\theta(z,A|x)\right] \tag{1}$$

$$= -\mathrm{KL}(q_\phi(z|A,x)||p_\theta(z)) + \mathbb{E}_{q_\phi(z|A,x)}\left[\log p_\theta(A|m,z)\right] \tag{2}$$

Here $q_\phi(z|A,x)$ is referred to as the encoder, parameterized by ϕ.

The prior over the latent variables can be set as the centered isotropic multivariate Gaussian $p_\theta(z) \sim \mathcal{N}(0,I)$ so that we can easily obtain samples, while $q_\phi(z|A,x)$ can be set as a multivariate Gaussian with a diagonal covariance $\mathcal{N}(\mu, \mathrm{diag}(\sigma))$ for simplicity. Both the encoder $q_\phi(z|A,x)$ and the decoder $p_\theta(A|x,z)$ can be implemented by neural networks (e.g. multi-layer perceptron MLP), in this case the mean vector μ and standard deviation vector σ are the outputs of the encoder.

Finally, in order to ensure that the model is conditioned on the background image x, in the encoder, we adopt a convolutional neural network to learn a visual vector representation, and concatenate such representation to the MLP-transformed A. In the decoder, we repeat the sampled vector so that it has the same width and height as the image x, then concatenate it with x along the channel dimension. We adopt another convolutional neural network to predict A^*, which is expected to be close to A using $L1$ distance. Figure 2 summarizes the architecture of our location module.

3.2 The Appearance Module

Once we obtain possible text locations from our location module, we can randomly pick stand-alone text instances and composite them on the background image. However, the composition results often look artificial and are not realistic enough. A text detector trained on such data often learns bias of the data distribution and cannot achieve good performance, as evidenced by Table 1.

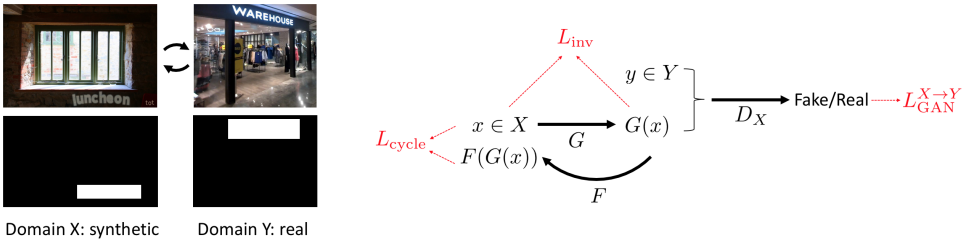

Figure 3: Architecture of the appearance module. It extends the Cycle-GAN [34] model by introducing an invariant loss L_{inv} to guide the generator to focus more on the objects highlighted by a mask.

Inspired by the recent Cycle-GAN [34] model that can translate image from one domain to another without paired training data, we proposed a Masked Cycle-GAN as our appearance module. It attempts to translate the synthetic images to real data domain. Here we first briefly describe the Cycle-GAN model, then explain how we utilize mask information to guide the model to focus more on the object of interest.

Cycle-GAN aims to learn mapping functions between two domains X and Y: $G : X \to Y$ and $F : Y \to X$. In order to make the distribution of image from $G(X)$ to be indistinguishable to that from Y (and vice versa), two discriminators D_X and D_Y are introduced such that the adversarial loss can be optimized. More specifically, for learning the mapping $G : X \to Y$, we can formulate the training objective as:

$$\arg\min_G \arg\max_{D_Y} L_{\text{GAN}}^{X \to Y} = \mathbb{E}_{y \sim p_{\text{data}(y)}} [\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}(x)}} [\log(1 - D_Y(G(x))] \quad (3)$$

Similarly, for learning the mapping $F : Y \to X$, the training objective can be formulated as:

$$\arg\min_G \arg\max_{D_X} L_{\text{GAN}}^{Y \to X} = \mathbb{E}_{x \sim p_{\text{data}(x)}} [\log D_X(x)] + \mathbb{E}_{y \sim p_{\text{data}(y)}} [\log(1 - D_X(F(y))] \quad (4)$$

However, such optimization is highly under-constrained. Due to the large capacity of the generators and discriminators, there may exist myriads of valid mappings. Not all of them can generate desired image translation results. To further constrain the learning process, a cycle-consistency loss is introduced:

$$L_{\text{cycle}} = ||F(G(x)) - x||_1 + ||G(F(y)) - y||_1 \quad (5)$$

In other words, L_{cycle} encourages that the translated image $G(x)$ can be further translated back to itself using F (and vice versa).

In this work, we want to translate images from synthetic data domain X to real data domain Y. Moreover, since we are able to obtain a mask for each image $x \sim X$ and $y \sim Y$ that highlights the location of text, we can utilize such information to better guide the generators. More specifically, we denote the mask as $M \in \{0,1\}^{W \times H}$, where W and H are the width and height of the image. Zeros indicate non-text region while ones indicate text region. Intuitively, we expect the appearance of the text region to be translated such that it is in harmony with the context, while the background is less changed. Therefore, we add an invariant loss:

$$L_{\text{inv}} = ||(1 - M_x) \odot G(x) - (1 - M_x) \odot x||_1 + ||(1 - M_y) \odot F(y) - (1 - M_y) \odot y||_1 \quad (6)$$

which provides an additional constrain on the generators. Here $\odot$ denotes pixel-wise masking operation.

The final training objective is: $L = L_{\text{GAN}}^{X\to Y} + L_{\text{GAN}}^{Y\to X} + \lambda_1 L_{\text{cycle}} + \lambda_2 L_{\text{inv}}$, where λ_1 and λ_2 are two hyper-parameters that control the reletive importance the corresponding terms.

4 Experiments

Here we evaluate the proposed text synthesis engine. We train a baseline text detection method on 1) real data (the training sets of ICDAR 2013 [14] and 2015 [15]), 2) our synthetic data and 3) the combination of the two above. Then we test on the test set of ICDAR 2015 dataset [15].

4.1 Datasets

ICDAR 2015 Incidental Scene Text dataset [15] is introduced in the ICDAR Robust Reading Competition for text detection. Images in this dataset are captured by a Google Glass "without taking actions to improve the positioning or quality of the text region". Therefore text in this dataset tends to be small, blurry or perspectively transformed. The training set contains 1,000 images with an average of 7.1 text regions per image. The test set contains 500 images.

We also use the training set of ICDAR 2013 Focused Scene Text dataset [14] during training. Images in this dataset are captured by a professional camera and they are deliberately focused on the text region. Consequently text in this dataset is often large, horizontal, and appears in the center of the image. On average each image has 4.2 text regions.

4.2 Baseline Method

We use EAST [33] as our baseline method for its simplicity and high performance. EAST is a fully-convolutional neural network. It first learns high-level and low-level representations via a encoder-decoder style network, then densely predict the text/non-text score, text box coordinates and text box rotation angle at each pixel location. The encoder-decoder backbone utilizes residual blocks to facilitate optimization, and skip connections to combine visual representations of different levels. More details of EAST can be found in [33].

4.3 Results

Table 2 shows of the performance of the baseline text detector (EAST) using different kinds of training images. From the table we can see that training only on *SynthText* significantly decreases the performance compared with training on real images, although the total amount of training images is almost 700 times larger. Such a huge gap in performance suggests that the previously proposed rule-based text synthesis engine may not capture the distribution of the real data. Moreover, even if we trained on the combination of real images (repeated 100 times so that synthetic data will not dominate the training set) and the *SynthText* images, the performance is still worse than only training on real images. Such fact suggests that the data distribution of *SynthText* is very different from the real data distribution.

As a comparison, using synthetic images that are generated by our learning-based text synthesis engine, the baseline EAST detector achieved 59.12 recall and 60.97 precision,

Train Set	#Train	Test Set	#Test	Recall	Precision	F1
IC13+15	1,229	IC15	500	77.76	84.35	81.18
SynthText	858,750	IC15	500	50.48	57.51	53.89
SynthText+100×(IC13+15)	981,650	IC15	500	74.88	83.39	78.91
Ours	858,750	IC15	500	59.12	60.97	60.03
Ours+100×(IC13+15)	981,650	IC15	500	**78.09**	**86.00**	**81.86**

Table 2: Performance of the baseline text detector (EAST [33]) using different kinds of training images. IC13 and IC15 are short for ICDAR 2013 and 2015 dataset, respectively.

Train Set	#Train	Test Set	#Test	Recall	Precision	F1
Ours-*app*	858,750	IC15	500	58.21	61.61	59.86
Ours-*app*+100×(IC13+15)	981,650	IC15	500	76.39	82.77	79.45
Ours	858,750	IC15	500	59.12	60.97	60.03
Ours+100×(IC13+15)	981,650	IC15	500	**78.09**	**86.00**	**81.86**

Table 3: Ablation study on the effectiveness of the appearance module. Ours-*app* denotes images that are generated only using our location module.

outperforming the one trained on *SynthText*. The performance is still worse than only training on real images, showing how difficult it is to generate "good" synthetic data for training. However, if we further add the real images (repeated 100 times so that synthetic data will not dominate the training set) to the training set, the baseline EAST detector achieved the best results: 78.09 recall and 86.00 precision, outperforming the one trained only on real images. Such results demonstrate the usefulness of our synthetic data.

To evaluate the effectiveness of our appearance module, we conduct an ablation study. Two kinds of synthetic images are generated: 1) images that are generated using both the location module and the appearance module, and 2) images that are generated using *only* the location module. The results are shown in Table 3. We can see that adding the appearance module consistently improve the performance.

4.4 Visualization Examples of the Location Module

Figure 4 shows some visualization examples of our location module. For each image, we sampled 50 latent vectors from the prior distribution, fed them into the decoder of our locations module and obtain the corresponding predictions, which are affine transformation parameters. The final sampled text locations can be obtained by transforming a reference box according to these affine transformations. From the visualization examples we can see that our location module tends to predict locations on the dominating object, such as the street sign, bus or wall. Sampled text locations are rarely observed in certain ares, such as sky or tree.

4.5 Visualization Examples of the Appearance Module

To better understand our appearance module, we visualized some images before and after the appearance translation. Examples are shown in Figure 5. We only shows the surrounding area of the text to better illustrate the difference. From these examples, we can see that

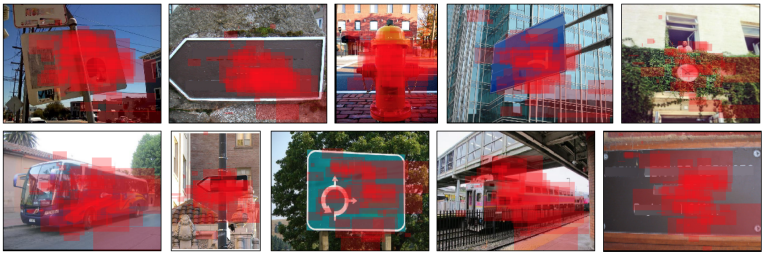

Figure 4: Sampled text locations from our location module.

Figure 5: Cropped image patches before (top row) and after (bottom row) our appearance module. We only shows the surrounding area of the text to better illustrate the differences.

our appearance module tends to add certain effects to the text-composited images, such as blurring or JPEG artifacts. This is an interesting finding, which may provide clues on how synthetic images are different from real images. Motion blurring, low resolution and JPEG artifacts are pervasive in real images, therefore it is desirable to have them during the data synthesis process.

5 Conclusions

We proposed the first learning-based and data-driven text synthesis engine. It first learns the conditional distribution of text locations on image plane using a variational auto-encoder, then composite text instances of the background image according to the locations sampled from the learned distribution. Then, a masked Cycle-GAN model is utilized to translate such text-composited images to ones that are indistinguishable from real-world images. We generated a large-scale synthetic dataset using our text synthesis engine and showed its effectiveness on the ICDAR 2015 dataset. When trained on our synthetic data, a baseline text detection method (EAST) outperforms one that is trained on previously proposed synthetic data. Combining our synthetic data and real data as the training set leads to the best performance. Future work includes learning text locations with perspective transformations and applying adversarial loss to the location module to generate more diversified samples.

Acknowledgements: We gratefully acknowledge partial support from NSF grant CCF 1317560 and a hardware grant from NVIDIA.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[2] Huizhong Chen, Sam S Tsai, Georg Schroth, David M Chen, Radek Grzeszczuk, and Bernd Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2609–2612. IEEE, 2011.

[3] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[4] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2963–2970. IEEE, 2010.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[7] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016.

[8] Dafang He, Xiao Yang, Wenyi Huang, Zihan Zhou, Daniel Kifer, and C Lee Giles. Aggregating local context for accurate scene text detection. In *Asian Conference on Computer Vision*, pages 280–296. Springer, 2016.

[9] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Deep direct regression for multi-oriented scene text detection. *arXiv preprint arXiv:1703.08289*, 2017.

[10] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

[11] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *European conference on computer vision*, pages 512–528. Springer, 2014.

[12] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2016.

[13] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2cnn: Rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017.

[14] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere de las Heras. Icdar 2013 robust reading competition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1484–1493. IEEE, 2013.

[15] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1156–1160. IEEE, 2015.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[17] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. In *Advances in Neural Information Processing Systems*, pages 10393–10403, 2018.

[18] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *arXiv preprint arXiv:1801.02765*, 2018.

[19] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. *arXiv preprint arXiv:1803.05265*, 2018.

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[21] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *arXiv preprint arXiv:1703.01086*, 2017.

[22] Yuchen Pu, Weiyao Wang, Ricardo Henao, Liqun Chen, Zhe Gan, Chunyuan Li, and Lawrence Carin. Adversarial symmetric variational autoencoder. In *Advances in Neural Information Processing Systems*, pages 4330–4339, 2017.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[24] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic back-propagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[25] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2550–2558, 2017.

[26] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *European Conference on Computer Vision*, pages 56–72. Springer, 2016.

[27] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.

[28] Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng. End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3304–3308. IEEE, 2012.

[29] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016.

[30] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C Lee Giles. Learning to read irregular text with attention mechanisms. In *IJCAI*, pages 3280–3286, 2017.

[31] Fangneng Zhan, Shijian Lu, and Chuhui Xue. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 249–266, 2018.

[32] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[33] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: An efficient and accurate scene text detector. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[35] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.