Enhancing Scene Text Detectors with Realistic Text Image Synthesis Using Diffusion Models

Ling Fu[a], Zijie Wu[b], Yingying Zhu[a,*], Yuliang Liu[b], Xiang Bai[b]

[a]School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, 430074, China
[b]School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, 430074, China

Abstract

Scene text detection techniques have garnered significant attention due to their wide-ranging applications. However, existing methods have a high demand for training data, and obtaining accurate human annotations is labor-intensive and time-consuming. As a solution, researchers have widely adopted synthetic text images as a complementary resource to real text images during pre-training. Yet there is still room for synthetic datasets to enhance the performance of scene text detectors. We contend that one main limitation of existing generation methods is the insufficient integration of foreground text with the background. To alleviate this problem, we present the **Diff**usion Model based **Text** Generator (**DiffText**), a pipeline that utilizes the diffusion model to seamlessly blend foreground text regions with the background's intrinsic features. Additionally, we propose two strategies to generate visually coherent text with fewer spelling errors. With fewer text instances, our produced text images consistently surpass other synthetic data in aiding text detectors. Extensive experiments on detecting horizontal, rotated, curved, and line-level texts demonstrate the effectiveness of DiffText in producing realistic text images. Code will be available at: https://github.com/99Franklin/DiffText.

Keywords: Scene text detection, Text image synthesis, Data augmentation

1. Introduction

Scene text detection techniques have seen significant advancements and attracted increased attention from researchers (Liu et al., 2019a; Long et al., 2021). These techniques have many practical applications in areas such as sign localization in the wild (Tsai et al., 2011), image retrieval based on scene text (Mishra et al., 2013) and assisting visually impaired users (Chessa et al., 2016). However, existing scene text detection methods require a large quantity of training data. The acquisition of sufficient scene text images and their accurate annotation are labor-intensive and time-consuming processes. To mitigate the above issues, researchers have proposed methods for generating synthetic text images (Gupta et al., 2016) and using them as pre-training data. Existing algorithms for generating text images can be roughly categorized into image composition methods and learning-based methods.

Image composition methods (Gupta et al., 2016; Liao et al., 2020a; Long and Yao, 2020; Yim et al., 2021; Zhan et al., 2018) aim to align digitally generated or user-provided text content with background images by optimizing surface smoothness in the combined images. Yet these methods sometimes present noticeable visual discrepancies between the foreground text and the background. These discrepancies mainly stem from challenges in blending different sources. Algorithms such as Poisson image editing (Pérez et al., 2003) used in SynthText (Gupta et al., 2016) and the planar meshes employed by UnrealText

(a) Synthetic samples by DiffText

(b) Synthetic samples by Poisson image editing

Figure 1: Some synthetic samples generated by DiffText and Poisson image editing (Pérez et al., 2003) methods.

(Long and Yao, 2020) mainly rely on visual features in a single image to optimize the blended results.

On the other hand, learning-based methods (Tang et al., 2022; Yang et al., 2019; Zhang et al., 2019) aim to capture the realistic appearance of real text through reference samples. By

*Corresponding author
Email address: yyzhu@hust.edu.cn (Yingying Zhu)

observing the additional reference data, these methods transform the generated text images to appear visually closer to real text images. Prior works (Yang et al., 2019; Zhang et al., 2019) focused on refining the blended results of text images through mapping learning between unpaired data. Another study (Tang et al., 2022) utilized text segmentation labels to precisely control the appearance learning process from reference images. These approaches mainly rely on learning the visual appearance of reference samples, while the background information from training samples is not fully leveraged.

To seamlessly blend visual text into background images, we leverage the diffusion model as an integration solution. Its robust capability to blend diverse sources makes it a suitable choice for incorporating visual text into scene images. Our pipeline, the Diffusion Model based Text Generator (DiffText), carries out a text-conditional image inpainting task, and can automatically produce realistic text images. Fig. 1 exhibits some synthetic samples. In the presented cases generated by Poisson image editing, the text contents appear to be pasted on the background image as watermarks. Our method allows the generated text to merge well with backgrounds, presenting superior integration effectiveness. DiffText consists of an autoencoder, a text encoder, and a denoising module. Given a natural image and a selected region for placing the visual text, we first mask the region to obtain a masked image that provides background information. This masked image is then projected into the latent space by the autoencoder, and the denoising module generates the foreground content from Gaussian noise by observing the data distribution of the background, conditioned on the input text string. The fusion of the foreground and background significantly enhances the visual fidelity of the synthesized text compared to previous methods.

To ensure the quality of the synthetic text images, we employ two strategies during the generation process. Firstly, we crop surrounding regions to obtain local backgrounds, facilitating efficient batch inpainting. Additionally, we utilize a pre-trained text recognizer to filter out low-quality instances. By incorporating these strategies into DiffText, we produced high-quality text images, highlighting the potential of our method in generating valuable synthetic data. Sufficient ablative experiments demonstrate the advantages of the DiffText.

We summarize the advantages of our methods as follows:

- To seamlessly integrate visual texts into background images, we present DiffText, a pipeline that uses the diffusion model to blend different sources, enabling the automatic production of realistic text images.

- To enhance the credibility of the generated visual text, we introduce two strategies that are incorporated into DiffText to assist in the generation process.

- We produced 10,000 scene text images by DiffText. With these images, we conducted comprehensive scene text detection experiments on detecting horizontal, rotated, curved, and line-level texts. The results present significant improvements in the performance of scene text detectors.

2. Related work

2.1. Scene text detection

Scene text detection technology, which aims to automatically localize text within a given image, has been an active research topic. Early text detectors primarily utilized hand-crafted features to search for text regions. For instance, STW (Epshtein et al., 2010) used edge detection to extract character candidates while MSER (Neumann and Matas, 2010, 2012) relied on extremal region extraction. Additionally, Yi and Tian (2013) and Minetto et al. (2014) respectively utilized character appearances and shape descriptors to help text detectors. In terms of modern scene text detectors, they can be divided into two categories, i.e., regression-based and segmentation-based methods.

Regression-based methods explicitly represent the shape of text instances by point sequences. TextBoxes (Liao et al., 2017) designed object anchors to fit the shape of text instances. Subsequently, EAST (Zhou et al., 2017) adopted pixel-level regression to deal with multi-oriented text instances in an anchor-free approach. To address the issue of Label Confusion, Liu et al. (2019c, 2021a) proposed to parameterize the bounding boxes into orderless sequences. Zhang et al. (2020) proposed to regress small rectangular components of text instances and use Graph Convolutional Network to model the connection of these components. To represent the contour of text instances more precisely, Fourier signatures were adopted by FCENet (Zhu et al., 2021). To detect texts under extreme traffic scenarios, He et al. (2023) proposed to transfer the text detection ability from conventional scenes to traffic scenes. With the emergence of DETR (Carion et al., 2020), DPText-DETR (Ye et al., 2023) leveraged learnable queries and attention mechanisms to model the point coordinates within the text region. As for segmentation-based methods, they typically generate the text proposals from pixel-level segmentation maps. Zhang et al. (2016) used semantic segmentation to predict text regions for multi-oriented instances. To perceive and distinguish adjacent text instances, PSENet (Wang et al., 2019) designed progressive scale expansion and Tian et al. (2019) introduced pixel embedding. To alleviate the issue of heavy post-processing, DBNet (Liao et al., 2020b) proposed a differentiable binarization module to perform the binarization process in the network. In our experimental analysis, we selected FCENet and DBNet, both of which excel in balancing performance and efficiency, from each category as text detectors.

2.2. Text image Generation

Text Image Generation involves creating visual texts on given natural images and has many applications in text analysis. Wang et al. (2012) proposed a character generator to synthesize training data for character-level text recognizers. To support word-level text recognition models, some methods focused on generating patch images (Jaderberg et al., 2016; Fogel et al., 2020; Kang et al., 2020; Yim et al., 2021; Nikolaidou et al., 2023; Zhu et al., 2023). In addition, other works concentrated on text content generation (Xie et al., 2021; Wang et al., 2022; Dai et al., 2023; Liu et al., 2023; Ma et al., 2023; Das et al.,

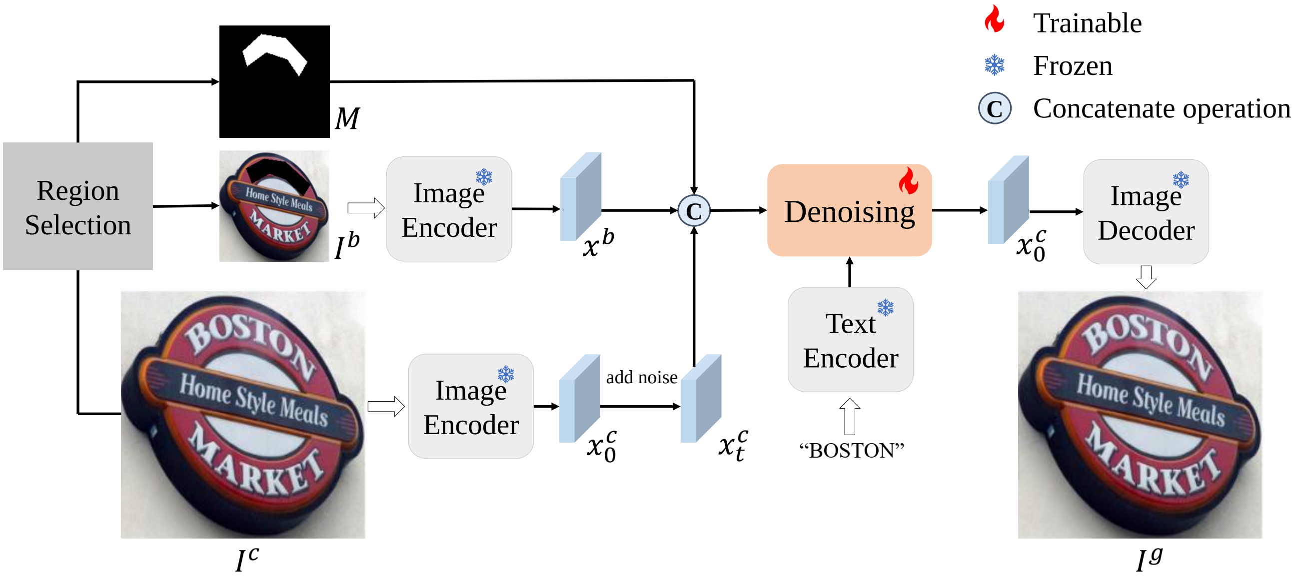


Figure 2: The overview framework of our proposed DiffText. Region Selection refers to the process of determining the placement region. Image Encoder and Image Decoder are from the autoencoder. Text Encoder is used to process the input textual string. Denoising refers to the denoising process to predict the added noise.

2022) or text image editing (Wu et al., 2019; Shimoda et al., 2021; Subramanian et al., 2021; Ji et al., 2023).

Besides, some studies explore inserting texts into scene images to aim for scene text detectors. Mainstream methods rely on image composition technologies, where digital text foregrounds are overlaid onto background images. Synthtext (Gupta et al., 2016) is the pioneering work that proposed to synthesize text images for scene text detectors. Poisson image editing (Pérez et al., 2003) was employed to blend foreground and background sources. Then VISD (Zhan et al., 2018) enhanced the harmonization of text instances using semantic information and a coloring scheme. SynthText3D (Liao et al., 2020a) and UnrealText (Long and Yao, 2020) leveraged 3D graphics engines to generate text images in 3D scenes. While CurveSynth (Liu et al., 2021b) aided in detecting arbitrary-shaped texts. Some learning-based methods (Tang et al., 2022; Yang et al., 2019; Zhang et al., 2019) focus on improving the realism of text appearance, but they face challenges due to the limited scope of reference data. In our work, we specifically focus on generating synthetic data that mixes visual texts with the backgrounds for the scene text detection task.

3. Methodology

The Diffusion Model based Text Generator (DiffText) is a pipeline designed to generate visually appealing text images. DiffText takes a background image and a selected region as input. It then executes a denoising process to transform Gaussian noise into visual content with the given text condition in a latent space. We will describe the details of DiffText and the generation strategies we use to produce synthetic data for scene text detection models in the following subsections.

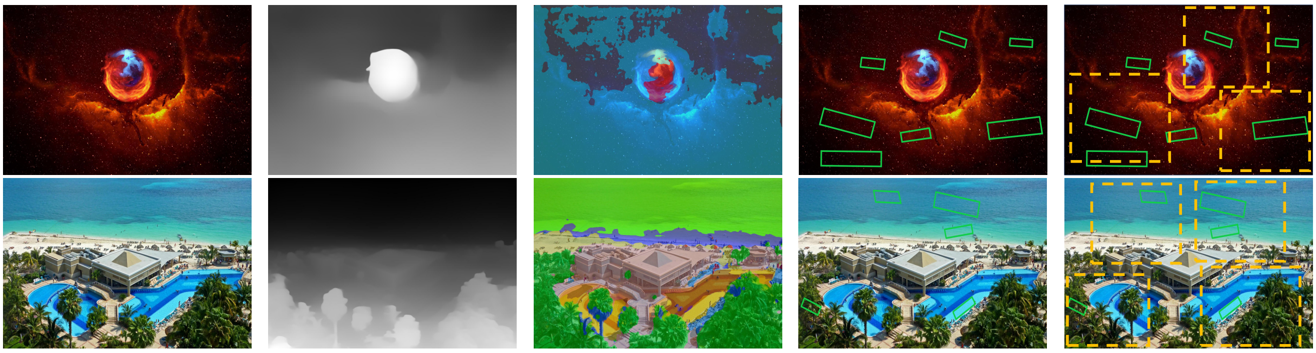
3.1. Synthetic pipeline

DiffText builds upon Stable Diffusion (Rombach et al., 2022), a method proposed to tackle the computational complexity issue in diffusion models (Sohl-Dickstein et al., 2015). The diffusion process involves adding noise to the samples, while the denoising process reconstructs the original sample from noised data. These processes are performed at the pixel level. Consequently, when the model's input is a high-resolution image, computational complexity becomes a significant challenge. To this end, Stable Diffusion proposed to encode the image into a latent space with an autoencoder, allowing the diffusion and denoising processes to occur in this space. Additionally, a condition encoder was employed to control the generative model's output under various conditions such as text, semantic maps, and others. More specifically, given an image x, the autoencoder translates it into a latent feature z_0. Gaussian noise is then added to z_0 to yield z_t at the current timestep t. Subsequently, a condition encoder τ_θ transforms the given condition y into the latent feature $\tau_\theta(y)$. The adopted squared error loss function for predicting noise can be defined as:

$$L = \mathbb{E}_{\epsilon(x),y,\epsilon\sim\mathcal{N}(0,1),t}[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2], \tag{1}$$

Here, ϵ denotes the actual added noise, while ϵ_θ represents the learnable parameters of the denoising module which predicts the added noise.

Our objective is to seamlessly integrate visual text into a natural image while maintaining background consistency to generate realistic text images. In addition to this, we aim to control the text content with the input string. Therefore, a text-conditional image inpainting approach is an ideal solution for these requirements. DiffText consists of three key components:



(a) Original images (b) Depth maps (c) Segmentation maps (d) Placement regions (e) Cropped regions

Figure 3: (a) The original background images. (b) The corresponding depth maps of the background images. (c) The segmentation maps of the background images. (d) The desired placement regions are indicated by green boxes. (e) The chosen cropped regions are represented by yellow boxes.

A VAE (Esser et al., 2021) to project image samples from pixel space to the latent space, a text encoder from Clip (Radford et al., 2021) to encode the input textual string. and a denoising module based on UNet (Ronneberger et al., 2015) to predict the noise to be added to the latent image feature during the diffusion process. The overview framework of DiffText is illustrated in Fig. 2.

To train DiffText, we start by collecting various public scene text image datasets. For each text instance, we mask the corresponding text region using the polygon label. This process creates paired training samples consisting of a masked image I^b and its corresponding original image I^c. Initially, I^c is projected into a latent feature x_0^c using the encoder of VAE, denoted as V_e. Gaussian noise ϵ is then iteratively added to the latent feature for t steps to transform it into a noised feature x_t^c. The process of obtaining x_t^c can be formulated as:

$$x_t^c = \sqrt{\overline{\alpha}_t} x_0^c + \sqrt{1-\overline{\alpha}_t}\epsilon, \quad \alpha_t = 1-\beta_t, \overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i, \tag{2}$$

where $\beta_i \in [0,1], \forall i \in [1,T]$. β is used to schedule the speed of adding noise, which is typically set to increase linearly. Additionally, the background region and mask information are provided to the generation model. To incorporate the additional information, I^b is also projected into the same latent space, resulting in a latent feature x^b that contains the background information. Then x_t^c, x^b, and a binary map M containing the mask information, are concatenated together. Besides, the desired text string is encoded into a text embedding t_e by the text encoder. Subsequently, the concatenated latent feature x_t^m and t_e are fed into the UNet within the denoising module. The cross-attention (Vaswani et al., 2017) layer is utilized to integrate t_e into the x_t^m. Finally, the recovered latent feature is passed through the decoder of VAE, producing the reconstructed image as the output.

During the training process, the parameters of the VAE and the text encoder are kept fixed, while the UNet is trainable. The optimization objective is defined as the squared error loss between the real Gaussian noise and the predicted noise. This can be formulated as:

Figure 4: Illustration of text instance filtering by a text recognizer. The generated text regions are evaluated by a pretrained text recognizer to determine their preservation.

$$L = \mathbb{E}_{x_t^m, t_e, \epsilon\sim\mathcal{N}(0,1), t}[||\epsilon - \epsilon_\theta(x_t^m, t, t_e)||_2^2], \tag{3}$$

In Eq. 3, ϵ represents the actual added noise, ϵ_θ represents the parameters of the UNet model, and t denotes the current timestep for prediction. During the inference phase, the latent feature x_t^c is substituted with Gaussian noise. Similar to the training phase, the masked image and the binary mask are still provided. Then, the denoising module generates the foreground content from the Gaussian noise, using the background information and the given text string. Finally, the decoder of VAE projects the latent feature into the pixel space to generate the text image.

3.2. Generation strategies

To generate visually coherent text, we put forth two strategies to aim for the generation process. Firstly, we design a local cropping approach to support forwarding DiffText in batch and reducing the noise during generating. To begin with, we follow the region selection method used in SynthText (Gupta et al., 2016). With the predicted depth information and segmentation maps. Later, we crop local regions from the background image that contain the desired text region(approximately four cropped regions per image). This allows us to forward the diffusion model in batch to expedite faster generation. Furthermore, this cropping operation helps to reduce noise brought to the background during the inpainting process and minimizes interference between the generated text instances. The entire process is illustrated in Fig. 3.

Secondly, we propose an instance filtering strategy to help generate credible text. Due to the inherent limitations in character-level content generation (Liu et al., 2023) for the diffusion model, we employ a pretrained text recognizer (Fang et al., 2021) to filter out instances with low confidence. Concretely, for the inpainting results of the aforementioned cropped regions, we extract the exact text instance region as a patch image. We then utilize the pretrained text recognizer to process these patch images. If the confidence score falls below a predefined threshold, the instance is discarded. The filtering process is depicted in Fig. 4. By selecting instances with high confidence scores, we ensure the quality of the generated text. Finally, we replace the cropped regions with the generated regions to finish text rendering.

4. Experiments

In this section, we provide detailed information about the training process of DiffText and the conducted scene text detection experiments. We begin by introducing the experimental setup, including text image datasets used for comparison, the generation process of our proposed synthetic dataset, as well as the implementation details of DiffText and the text detectors. Following that, we give a comprehensive analysis of the scene text detection experiments.

4.1. Experimental setup

4.1.1. Datasets

To demonstrate the effectiveness of DiffText, we performed a comprehensive analysis comparing the text images generated by DiffText with previous synthetic datasets, including SynthText (Gupta et al., 2016), VISD (Zhan et al., 2018), CurveSynth (Liu et al., 2021b), SynthText3D (Liao et al., 2020a), and UnrealText (Long and Yao, 2020). Additionally, we used four real text datasets, encompassing ICDAR2013 (IC13) (Karatzas et al., 2013), ICDAR2015 (IC15) (Karatzas et al., 2013), CTW1500 (CTW) (Liu et al., 2019b), and TotalText (Ch'ng et al., 2020), as test data to evaluate the performance of our approach. In the following paragraphs, we provide a brief introduction to these datasets.

- SynthText is a widely adopted solution for generating synthetic data to support scene text detection and recognition models. It serves as a valuable resource during the pretraining stage of these algorithms.
- VISD enhanced the synthetic effect of SynthText by leveraging semantic segmentation information and the proposed color scheme. This results in the generated text appearing more visually harmonious.

Table 1: Calculation of the number of text instances in each synthetic dataset.

Synthetic data	Text instances
SynthText 10K	84885
VISD 10K	94359
CurveSynth 10K	122405
SynthText3D 10K	143102
UnrealText 10K	268394
DiffText 10K	76354

- CurveSynth complements SynthText by providing numerous extra arbitrary-shaped texts. While SynthText mainly focused on generating rotating rectangles, CurveSynth introduced a broader range of text shapes.
- SynthText3D utilized camera views from Unreal Engine 4, and UnrealCV as the background source. Synthetic digital texts were then rendered onto these scenes to simulate real-world scenarios.
- UnrealText is another notable work that employs game engines to generate complex scenes for synthetic text. It utilized collision detection techniques to automatically find suitable placements for the text within the scene.
- IC13 is a scene text dataset specifically curated for text detection and recognition tasks. In this dataset, the text instances are predominantly horizontally oriented and located at the center of the image.
- IC15 is another dataset commonly used for text detection and recognition. A notable distinction is that it contains numerous small and blurred text instances, and the text instances are mainly in the shape of rotated rectangles. This poses a significant challenge for the algorithms.
- CTW contains many line-level instances with arbitrary shapes. It is widely used as a benchmark to evaluate the performance of text detectors, particularly in detecting line-level texts and curved texts.
- TotalText is another scene text dataset that stands out for containing many arbitrary shape text. This dataset is specifically designed to address the scenario of curved texts, providing valuable resources for the community.

4.1.2. DiffText 10K

By leveraging the generation capabilities of DiffText, we created 10,000 high-quality text images called DiffText 10K. To ensure a fair comparison with previous synthetic datasets, we randomly selected 10,000 images from SynthText, VISD, CurveSynth, SynthText3D, and UnrealText. These subsets are denoted as SynthText 10K, VISD 10K, CurveSynth 10K, SynthText3D 10K, and UnrealText 10K, and are used in the subsequent comparison experiments. The background images used in our synthetic dataset remain consistent with those from the SynthText dataset, while the text instances are generated using our proposed strategy. The quantities of text instances contained

Table 2: Scene text detection results of DBNet (Liao et al., 2020b) models trained solely on each synthetic dataset, and tested on each real text dataset without fine-tuning.

Training data	IC13			IC15			Totaltext		
	Precision	Recall	Hmean	Precision	Recall	Hmean	Precision	Recall	Hmean
SynthText 10K	74.77	60.64	66.97	59.02	44.25	50.58	58.61	37.34	45.62
VISD 10K	80.15	67.85	73.49	67.12	47.57	55.68	61.88	41.85	49.93
CurveSynth 10K	74.29	54.89	63.13	64.13	28.41	39.37	61.61	31.74	41.90
SynthText3D 10K	84.62	66.85	74.69	69.53	51.85	59.40	56.30	40.72	47.26
UnrealText 10K	81.94	65.48	72.79	67.99	43.86	53.32	48.80	29.39	36.69
DiffText 10K	83.88	72.24	**77.63**	79.32	49.49	**60.95**	65.65	44.60	**53.12**

Table 3: Scene text detection results of FCENet (Zhu et al., 2021) models trained solely on each synthetic dataset, and tested on each real text dataset without fine-tuning.

Training data	IC13			IC15			Totaltext		
	Precision	Recall	Hmean	Precision	Recall	Hmean	Precision	Recall	Hmean
SynthText 10K	54.59	52.69	53.62	67.67	50.79	58.03	51.84	49.53	50.66
VISD 10K	34.08	48.58	40.06	75.15	58.98	66.09	48.68	47.54	48.10
CurveSynth 10K	55.80	48.31	51.79	61.37	36.78	46.00	50.47	40.99	45.24
SynthText3D 10K	61.84	43.65	51.18	70.06	61.29	65.38	46.33	42.98	44.59
UnrealText 10K	68.87	50.32	58.15	74.97	57.82	65.29	54.58	44.15	48.81
DiffText 10K	57.96	64.47	**61.05**	76.77	60.13	**67.44**	57.12	55.40	**56.25**

in these synthetic datasets are listed in Tab. 1. Due to our instance filtering strategy, the number of samples in DiffText 10K is slightly lower than that of SynthText 10K. The generation process was executed on an RTX 3090 GPU and took approximately 30 hours.

4.1.3. Implementation details

In all experiments, we utilized PyTorch to implement our models. To train the DiffText, we collected some public scene text datasets, including IC13 (Karatzas et al., 2013), IC15 (Karatzas et al., 2015), ICDAR2019 (Nayef et al., 2019), TotalText (Ch'ng et al., 2020), COCO-Text (Veit et al., 2016), CTW1500 (Liu et al., 2019b), ArT (Chng et al., 2019), LSVT (Sun et al., 2019) and TextOCR (Singh et al., 2021). The combined total number of text instances in these datasets is 520,337. Each instance is processed by masking the text region and forming a paired sample with the original image for training. To leverage the strong generation capabilities of the Stable Diffusion (Rombach et al., 2022)[1], we initialize the VAE (Esser et al., 2021), the text encoder of CLIP (Radford et al., 2021), and the UNet (Ronneberger et al., 2015) by loading the parameters of its 2.0-base model. During training, VAE and the text encoder are frozen, while the parameters of UNet are learnable. We utilize the AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 1e-5, $\beta 1$=0.9, $\beta 2$=0.999, and a weight decay of 1e-2. The batch size is set to 24, and the total training epoch is 20. The input images are resized to 512 pixels.

For the scene text detection experiment, we employ two popular text detectors, a segmentation-based text detector (DBNet) (Liao et al., 2020b) and a regression-based text detector

Table 4: Scene text detection results of DBNet (Liao et al., 2020b) models trained on each synthetic dataset and IC15 (Karatzas et al., 2015). The DBNet models are initially trained solely on each synthetic dataset, followed by fine-tuning on IC15, and then tested on IC15. † denotes that the DBNet models utilize the parameters of ResNet-50 (He et al., 2016) trained on ImageNet (Deng et al., 2009).

Training data	IC15		
	Precision	Recall	Hmean
IC15	87.10	75.73	81.02
IC15†	87.20	81.66	84.34
IC15 + SynthText 10K	88.38	83.10	85.66
IC15 + VISD 10K	88.68	82.57	85.51
IC15 + CurveSynth 10K	89.33	82.62	85.84
IC15 + SynthText3D 10K	87.72	82.52	85.04
IC15 + UnrealText 10K	88.76	82.86	85.71
IC15 + DiffText 10K	89.79	83.44	**86.50**

(FCENet) (Zhu et al., 2021). The training strategies follow the default configuration of mmocr's implementation[2]. Specifically, for DBNet, the pretraining procedure adopts Stochastic gradient descent (SGD) as the optimizer with a learning rate of 0.007, momentum of 0.9, and a weight decay of 1e-4 for training 100,000 iterations. During the fine-tuning procedure, the model is trained for 1200 epochs. For FCENet, in the pretraining procedure, the learning rate is set to 0.001, weight decay is set to 5e-4, and the other settings remain the same as those in DBNet. All experiments are conducted on RTX 3090 GPUs.

[1]https://github.com/huggingface/diffusers

[2]https://github.com/open-mmlab/mmocr

(a) Samples from DiffText 10K

(b) Samples from SynthText 10K

(c) Samples from VISD 10K

(d) Samples from CurveSynth 10K

(e) Samples from SynthText3D 10K

(f) Samples from UnrealText 10K

Figure 5: Some samples from each synthetic dataset. (a) Samples from the DiffText; (b) samples from the SynthText (Gupta et al., 2016); (c) samples from VISD (Zhan et al., 2018); (d) samples from CurveSynth (Liu et al., 2021b); (e) samples from SynthText3D (Liao et al., 2020a); (f) samples from UnrealText (Long and Yao, 2020).

Table 5: Scene text detection results of DBNet (Liao et al., 2020b) models trained on each synthetic dataset and CTW (Liu et al., 2019b) (short for SCUT-CTW1500). The DBNet models are initially trained solely on each synthetic dataset, followed by fine-tuning on CTW, and then tested on CTW. † denotes that the DBNet models utilize the parameters of ResNet-50 (He et al., 2016) trained on ImageNet (Deng et al., 2009).

Training data	CTW		
	Precision	Recall	Hmean
CTW	71.25	65.29	68.14
CTW†	72.00	69.27	70.61
CTW + SynthText 10K	75.79	68.01	71.69
CTW + VISD 10K	74.67	68.01	71.18
CTW + CurveSynth 10K	74.66	69.94	72.22
CTW + SynthText3D 10K	74.32	68.90	71.51
CTW + UnrealText 10K	74.49	69.20	71.75
CTW + DiffText 10K	76.31	71.06	**73.59**

4.2. Comparison with previous methods

Only synthetic data. We trained two popular text detectors, DBNet (Liao et al., 2020b) and FCENet (Zhu et al., 2021), with each synthetic text dataset to demonstrate the effectiveness of DiffText. The experiment results are presented in Tab. 2 and Tab. 3. From the tables, we observe that the text detectors trained on synthetic data perform well on real text datasets. This demonstrates the necessity of generating synthetic data, especially when high-quality text annotations are scarce. Furthermore, the text detector trained on DiffText 10K consistently outperforms other text detectors on all datasets, validating the realism of text in DiffText 10K and the effective density estimation capability of DiffText.

We present samples from DiffText 10K along with other synthetic datasets in Fig. 5. Due to the limitations in blending foreground texts and background, the scene texts in SynthText (Gupta et al., 2016), VISD (Zhan et al., 2018) and CurveSynth (Liu et al., 2021b) visually appear to be pasted onto the background images. On the other hand, SynthText3D (Liao et al., 2020a) and UnrealText (Long and Yao, 2020) exhibit better harmony between the texts and the backgrounds. This is mainly because both the texts and the backgrounds in these datasets are digitally simulated, resulting in a more cohesive visual appearance. However, there still exists an inevitable domain gap between the simulated scenes and real images. In contrast, DiffText overcomes these limitations by learning the data distribu-

Table 6: Scene text detection results of DBNet (Liao et al., 2020b) trained on different synthetic data under various setting. "With crop" refers to the generation process with region crop strategy, "With rec" indicates the generation process with instances filtering strategy.

With crop	With rec	IC13			IC15			Totaltext		
		Precision	Recall	Hmean	Precision	Recall	Hmean	Precision	Recall	Hmean
		81.97	61.46	70.25	59.28	39.53	47.43	62.32	41.44	49.78
✓		78.48	64.29	70.68	69.19	41.41	51.81	59.99	42.57	49.80
	✓	82.23	65.94	73.19	67.87	39.96	50.30	67.09	43.34	52.66
✓	✓	83.88	72.24	**77.63**	79.32	49.49	**60.95**	65.65	44.60	**53.12**

tion of entire real text images and seamlessly integrating text regions into the background. This approach ensures that our generated texts appear more realistic and provide greater benefits to text detectors. It is noteworthy that the number of text instances in DiffText 10K is significantly lower compared to other methods, as described in Tab.1, which supports our claim.

Synthetic data and real data. To further demonstrate the advantages of our generated text images, we conducted experiments by training text detectors using a combination of real and synthetic data. Initially, the DBNet (Liao et al., 2020b) models were trained on synthetic data, followed by fine-tuning on real text images. The results of these experiments are presented in Tab. 4. Through pretraining on synthetic data, the text detectors significantly improved their performance on the test datasets. The gains achieved by incorporating 10K synthetic data into the text detectors surpassed the improvements obtained from well-trained parameters sourced from ImageNet (Deng et al., 2009). This highlights the necessity of utilizing synthetic data for pre-training. Notably, due to the realistic nature of the text in DiffText 10K, the corresponding text detectors inherited enhanced visual text perception abilities during pre-training. Consequently, DiffText 10K provided better benefits for the text detectors compared to other synthetic text datasets.

Moreover, we replace the real data of the previous experiment from IC15 to CTW1500 and replicate the fine-tuning process to evaluate the performance on line-level texts and curved texts. The results are displayed in Tab. 5. Despite the lack of curved text instances, DiffText 10K still presents superior performance. This indicates that our text images can effectively improve the robust generalization ability of the text detector.

4.3. Ablation studies

To evaluate the effectiveness of our proposed generation strategies, we conducted ablation studies. Synthetic data was generated under various settings and used as training data for the DBNet (Liao et al., 2020b) text detectors. The results on each real text dataset are presented in Tab. 6. "With Crop" refers to the generation process utilizing the region crop strategy, while "With Rec" indicates the generation process involving instances filtering strategy with text recognizer.

From the table, it is evident that the detection performance significantly drops without the region crop strategy or instances filtering strategy. As mentioned earlier, the region crop strategy helps the model reduce noise introduced by the inpainting process and diminish interference among generated text instances. On the other hand, the instances filtering strategy mitigates the issue of generating unstable text, thereby enhancing the production of discriminative text instances.

5. Discussion

Our studies establish the necessity of synthetic data for training scene text detectors. Furthermore, our produced high-quality text images yield greater benefits. Compared to previous methods, DiffText enhances the realism of visual text by seamlessly integrating the foreground content with the background. In future studies, more advanced generation models will produce more legible text content, thereby improving the performance of scene text spotters and helping many other downstream applications. Additionally, the efficiency of the generation process could be further optimized to expedite the production of large amounts of text images, which can support the further advancement of scene text detection technologies.

6. Conclusion

In this paper, we present DiffText, a pipeline that seamlessly integrates foreground text into background images to generate realistic scene text images. DiffText treats text image generation as a text-conditional image inpainting task. Specifically, it fulfills the foreground content by observing the intrinsic features of the background region and the text condition. Using DiffText, we produced a collection of high-quality text images, highlighting the potential of DiffText in generating valuable text datasets. Furthermore, we conducted comprehensive scene text detection experiments to demonstrate the realism of these text images. Experiment results show that our generated text images exhibit a diminished domain gap compared to previous synthetic data, resulting in significant benefits for scene text detectors.

References

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer. pp. 213–229.

Chessa, M., Noceti, N., Odone, F., Solari, F., Sosa-García, J., Zini, L., 2016. An integrated artificial vision framework for assisting visually impaired users. Computer Vision and Image Understanding 149, 209–228.

Chng, C.K., Liu, Y., Sun, Y., Ng, C.C., Luo, C., Ni, Z., Fang, C., Zhang, S., Han, J., Ding, E., et al., 2019. Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art, in: 2019 International Conference on Document Analysis and Recognition, IEEE. pp. 1571–1576.

Ch'ng, C.K., Chan, C.S., Liu, C.L., 2020. Total-text: toward orientation robustness in scene text detection. International Journal on Document Analysis and Recognition 23, 31–52.

Dai, G., Zhang, Y., Wang, Q., Du, Q., Yu, Z., Liu, Z., Huang, S., 2023. Disentangling writer and character styles for handwriting generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5977–5986.

Das, A., Yang, Y., Hospedales, T., Xiang, T., Song, Y.Z., 2022. Chirodiff: Modelling chirographic data with diffusion models, in: The Eleventh International Conference on Learning Representations.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 248–255.

Epshtein, B., Ofek, E., Wexler, Y., 2010. Detecting text in natural scenes with stroke width transform, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 2963–2970.

Esser, P., Rombach, R., Ommer, B., 2021. Taming transformers for high-resolution image synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12873–12883.

Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y., 2021. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7098–7107.

Fogel, S., Averbuch-Elor, H., Cohen, S., Mazor, S., Litman, R., 2020. Scrabblegan: Semi-supervised varying length handwritten text generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4324–4333.

Gupta, A., Vedaldi, A., Zisserman, A., 2016. Synthetic data for text localisation in natural images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2315–2324.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE. pp. 770–778.

He, X., Li, Z., Lin, J., Nai, K., Yuan, J., Li, Y., Wang, R., 2023. Domain adaptive multigranularity proposal network for text detection under extreme traffic scenes. Computer Vision and Image Understanding 233, 103709.

Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A., 2016. Reading text in the wild with convolutional neural networks. International Journal of Computer Vision 116, 1–20.

Ji, J., Zhang, G., Wang, Z., Hou, B., Zhang, Z., Price, B., Chang, S., 2023. Improving diffusion models for scene text editing with dual encoders. arXiv preprint arXiv:2304.05568 .

Kang, L., Riba, P., Wang, Y., Rusiñol, M., Fornés, A., Villegas, M., 2020. Ganwriting: Content-conditioned generation of styled handwritten word images, in: European Conference on Computer Vision, pp. 273–289.

Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al., 2015. Icdar 2015 competition on robust reading, in: 13th International Conference on Document Analysis and Recognition, IEEE. pp. 1156–1160.

Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P., 2013. Icdar 2013 robust reading competition, in: 12th International Conference on Document Analysis and Recognition, IEEE. pp. 1484–1493.

Liao, M., Shi, B., Bai, X., Wang, X., Liu, W., 2017. Textboxes: a fast text detector with a single deep neural network, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 4161–4167.

Liao, M., Song, B., Long, S., He, M., Yao, C., Bai, X., 2020a. Synthtext3d: synthesizing scene text images from 3d virtual worlds. Science China Information Sciences 63, 1–14.

Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X., 2020b. Real-time scene text detection with differentiable binarization, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 11474–11481.

Liu, R., Garrette, D., Saharia, C., Chan, W., Roberts, A., Narang, S., Blok, I., Mical, R., Norouzi, M., Constant, N., 2023. Character-aware models improve visual text rendering, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics. pp. 16270–16297.

Liu, X., Meng, G., Pan, C., 2019a. Scene text detection and recognition with advances in deep learning: a survey. International Journal on Document Analysis and Recognition 22, 143–162.

Liu, Y., He, T., Chen, H., Wang, X., Luo, C., Zhang, S., Shen, C., Jin, L., 2021a. Exploring the capacity of an orderless box discretization network for multi-orientation scene text detection. International Journal of Computer Vision 129, 1972–1992.

Liu, Y., Jin, L., Zhang, S., Luo, C., Zhang, S., 2019b. Curved scene text detection via transverse and longitudinal sequence connection. Pattern Recognition 90, 337–345.

Liu, Y., Shen, C., Jin, L., He, T., Chen, P., Liu, C., Chen, H., 2021b. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, 8048–8064.

Liu, Y., Zhang, S., Jin, L., Xie, L., Wu, Y., Wang, Z., 2019c. Omnidirectional scene text detection with sequential-free box discretization, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 3052–3058.

Long, S., He, X., Yao, C., 2021. Scene text detection and recognition: The deep learning era. International Journal of Computer Vision 129, 161–184.

Long, S., Yao, C., 2020. Unrealtext: Synthesizing realistic scene text images from the unreal world. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition , 5488–5497.

Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization, in: 7th International Conference on Learning Representations.

Ma, J., Zhao, M., Chen, C., Wang, R., Niu, D., Lu, H., Lin, X., 2023. Glyphdraw: Learning to draw chinese characters in image synthesis models coherently. arXiv preprint arXiv:2303.17870 .

Minetto, R., Thome, N., Cord, M., Leite, N.J., Stolfi, J., 2014. Snoopertext: A text detection system for automatic indexing of urban scenes. Computer Vision and Image Understanding 122, 92–104.

Mishra, A., Alahari, K., Jawahar, C., 2013. Image retrieval using textual cues, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 3040–3047.

Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.C., Liu, C.l., et al., 2019. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019, in: 2019 International Conference on Document Analysis and Recognition, IEEE. pp. 1582–1587.

Neumann, L., Matas, J., 2010. A method for text localization and recognition in real-world images, in: Proceedings of the 10th Asian Conference on Computer Vision-Volume Part III, pp. 770–783.

Neumann, L., Matas, J., 2012. Real-time scene text localization and recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 3538–3545.

Nikolaidou, K., Retsinas, G., Christlein, V., Seuret, M., Sfikas, G., Smith, E.B., Mokayed, H., Liwicki, M., 2023. Wordstylist: Styled verbatim handwritten text generation with latent diffusion models, in: Document Analysis and Recognition - 2023 - 17th International Conference, San José, CA, USA, August 21-26, 2023, Proceedings, Part II, Springer. pp. 384–401.

Pérez, P., Gangnet, M., Blake, A., 2003. Poisson image editing. ACM Transactions on Graphics 22, 313–318.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., 2021. Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR. pp. 8748–8763.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer. pp. 234–241.

Shimoda, W., Haraguchi, D., Uchida, S., Yamaguchi, K., 2021. De-rendering stylized texts, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1076–1085.

Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., Hassner, T., 2021. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8802–8812.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S., 2015. Deep unsupervised learning using nonequilibrium thermodynamics, in: International Conference on Machine Learning, PMLR. pp. 2256–2265.

Subramanian, J., Chordia, V., Bart, E., Fang, S., Guan, K., Bala, R., et al., 2021.

Strive: Scene text replacement in videos, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14549–14558.

Sun, Y., Ni, Z., Chng, C.K., Liu, Y., Luo, C., Ng, C.C., Han, J., Ding, E., Liu, J., Karatzas, D., et al., 2019. Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt, in: 2019 International Conference on Document Analysis and Recognition, IEEE. pp. 1557–1562.

Tang, Z., Miyazaki, T., Omachi, S., 2022. A scene-text synthesis engine achieved through learning from decomposed real-world data. arXiv preprint arXiv:2209.02397 .

Tian, Z., Shu, M., Lyu, P., Li, R., Zhou, C., Shen, X., Jia, J., 2019. Learning shape-aware embedding for scene text detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4234–4243.

Tsai, S.S., Chen, H., Chen, D., Schroth, G., Grzeszczuk, R., Girod, B., 2011. Mobile visual search on printed documents using text and low bit-rate features, in: 2011 18th IEEE International Conference on Image Processing, IEEE. pp. 2601–2604.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in Neural Information Processing Systems 30.

Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S., 2016. Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 .

Wang, T., Wu, D.J., Coates, A., Ng, A.Y., 2012. End-to-end text recognition with convolutional neural networks, in: Proceedings of the 21st International Conference on Pattern Recognition, IEEE. pp. 3304–3308.

Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S., 2019. Shape robust text detection with progressive scale expansion network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9336–9345.

Wang, Y., Pu, G., Luo, W., Wang, Y., Xiong, P., Kang, H., Lian, Z., 2022. Aesthetic text logo synthesis via content-aware layout inferring, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2436–2445.

Wu, L., Zhang, C., Liu, J., Han, J., Liu, J., Ding, E., Bai, X., 2019. Editing text in the wild, in: Proceedings of the 27th ACM International Conference on Multimedia, pp. 1500–1508.

Xie, Y., Chen, X., Sun, L., Lu, Y., 2021. Dg-font: Deformable generative networks for unsupervised font generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5130–5140.

Yang, X., He, D., Kifer, D., Giles, C.L., 2019. A learning-based text synthesis engine for scene text detection., in: British Machine Vision Conference, p. 94.

Ye, M., Zhang, J., Zhao, S., Liu, J., Du, B., Tao, D., 2023. Dptext-detr: Towards better scene text detection with dynamic points in transformer, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 3241–3249.

Yi, C., Tian, Y., 2013. Text extraction from scene images by character appearance and structure modeling. Computer Vision and Image Understanding 117, 182–194.

Yim, M., Kim, Y., Cho, H.C., Park, S., 2021. Synthtiger: Synthetic text image generator towards better text recognition models, in: International Conference on Document Analysis and Recognition, Springer. pp. 109–124.

Zhan, F., Lu, S., Xue, C., 2018. Verisimilar image synthesis for accurate detection and recognition of texts in scenes, in: Proceedings of the European Conference on Computer Vision, pp. 249–266.

Zhang, C., Zhan, F., Zhu, H., Lu, S., 2019. Scene text synthesis for efficient and effective deep network training. arXiv preprint arXiv:1901.09193 .

Zhang, S.X., Zhu, X., Hou, J.B., Liu, C., Yang, C., Wang, H., Yin, X.C., 2020. Deep relational reasoning graph network for arbitrary shape text detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9699–9708.

Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X., 2016. Multi-oriented text detection with fully convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4159–4167.

Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J., 2017. East: an efficient and accurate scene text detector, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5551–5560.

Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L., Zhang, W., 2021. Fourier contour embedding for arbitrary-shaped text detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3123–3131.

Zhu, Y., Li, Z., Wang, T., He, M., Yao, C., 2023. Conditional text image generation with diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14235–14245.