Learning to Draw Text in Natural Images with Conditional Adversarial Networks

Shancheng Fang[1,2] , **Hongtao Xie**[3*] , **Jianjun Chen**[1] , **Jianlong Tan**[1] , **Yongdong Zhang**[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]School of Information Science and Technology, University of Science and Technology of China
{fangshancheng,chenjianjun,tanjianlong}@iie.ac.cn, {htxie,zhyd73}@ustc.edu.cn,

Abstract

In this work, we propose an entirely learning-based method to automatically synthesize text sequence in natural images leveraging conditional adversarial networks. As vanilla GANs are clumsy to capture structural text patterns, directly employing GANs for text image synthesis typically results in illegible images. Therefore, we design a two-stage architecture to generate repeated characters in images. Firstly, a character generator attempts to synthesize local character appearance independently, so that the legible characters in sequence can be obtained. To achieve style consistency of characters, we propose a novel style loss based on variance-minimization. Secondly, we design a pixel-manipulation word generator constrained by self-regularization, which learns to convert local characters to plausible word image. Experiments on SVHN dataset and ICDAR, IIIT5K datasets demonstrate our method is able to synthesize visually appealing text images. Besides, we also show the high-quality images synthesized by our method can be used to boost the performance of a scene text recognition algorithm.

1 Introduction

Drawing text can be considered as a problem of image synthesis that focuses on text rendering. We humans are able to write or design text, practicing in communication, typography, hand-lettering, etc. However, it remains a challenge for artificial intelligence to manipulate the probability distributions of text images in uncontrolled conditions. In this work, we explore an approach to automatically synthesize text sequence in natural images given the text labels.

A straightforward way to synthesize text images is applying the non-learning based method [Jaderberg *et al.*, 2014; Gupta *et al.*, 2016; Zhan *et al.*, 2018]. By setting several parameters in advance, such as font, color, distortion, noise, etc. [Jaderberg *et al.*, 2014], this kind of method has the ability to semi-automatically render images with diverse text using the techniques of computer graphics. However, it is tedious for a non-learning approach to appropriately select the parameters and realistically simulate natural images (uneven lighting, occlusion, degradation, etc.). In addition, it also fails to synthesize images with unseen font, e.g., hand-writing and special-effects text.

To better simulate natural scenes and produce realistic text images, the learning based method can be considered. Recently, image synthesis has advanced dramatically with the emergence of generative adversarial networks (GANs) [Goodfellow *et al.*, 2014]. Though GANs show impressive results in high-quality image generation, it is still an intractable problem to capture geometric or structural scenes [Zhang *et al.*, 2018; Ledig *et al.*, 2017]. As characters occur repeatedly in text string, there are strong structural patterns in text images. Therefore, directly employing vanilla GANs for text image synthesis typically recovers unreadable or meaningless images (Figure 7a). Specifically, applying learning based method for text image synthesis confronts the following challenges: firstly, the text images should be generated in arbitrary length. Secondly, compared to the image-to-image translation task [Isola *et al.*, 2017] that requires an image as input, synthesizing text images from scratch only takes character labels, which carry less information than images. Besides, different from object synthesis that generally one salient category is placed in the images, characters in text sequence are of the same importance. Thus, how to keep all characters readable is the primary issue. In addition, characters possess dependency with each other and tend to share similar styles in natural images.

Taking the above challenges into consideration, we propose an entirely learning-based method leveraging conditional adversarial networks, called scene text synthesis GAN (STS-GAN). Inspired by the learning process that humans master characters writing first, and then are capable of drawing text freely, we divide the generation procedure into character synthesis and word synthesis stages. Figure 1 illustrates the framework of STS-GAN. In the former stage, a character generator attempts to capture local character appearance based on conditional GAN. The character generator separately synthesizes character images, so that the legible characters in sequence can be obtained. To resolve the problem of style discrepancy, we propose a novel style loss based on variance-minimization. The style loss aims to minimize the characters distance in style space, enabling char-

* Hongtao Xie (htxie@ustc.edu.cn)

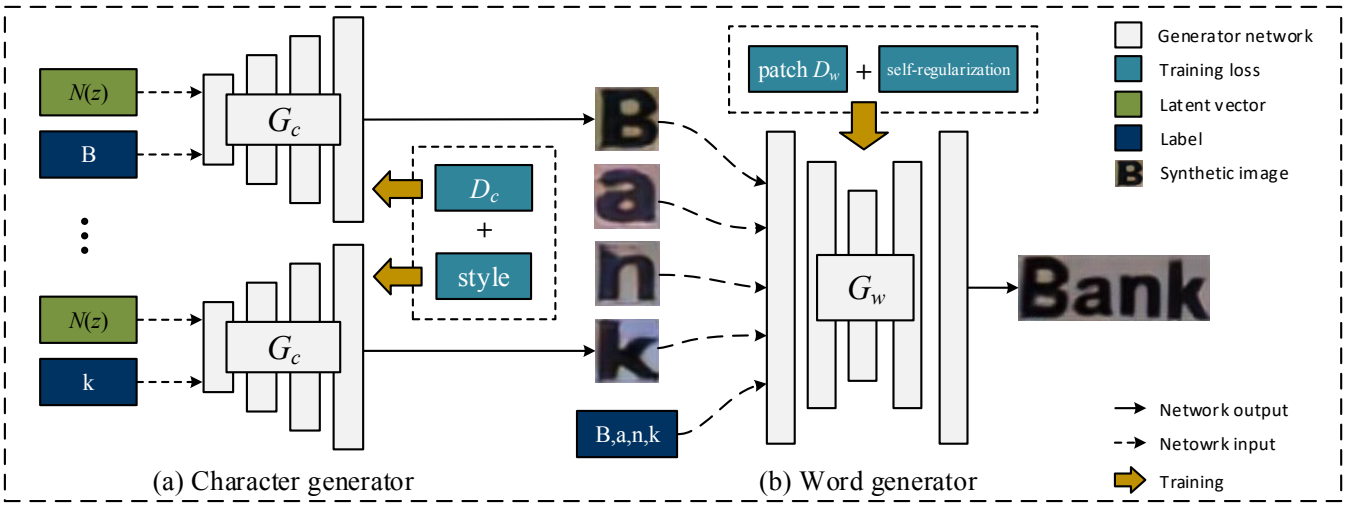


Figure 1: Overview of our approach. (a) Taking character labels and latent vector as input, character generator synthesizes local character images, which is trained with adversarial loss and style loss. (b) The synthetic character images are then fed into word generator conditioned on text labels, and global word image is synthesized through pixel-wise manipulation. The word generator is trained with patch level adversarial loss and $L1$ self-regularization loss.

acters in the same text patch to share similar styles (e.g., font and background). In the latter stage, we design a pixel-manipulation word generator conditioned on text labels to produce variable-length text images. Training with adversarial learning and self-regularization, the word generator is able to learn the characters dependency in natural scenes, and it can consequently map the synthetic character images to perceptually convincing word image. To the best of our knowledge, our method is the first one to synthesize text images from scratch entirely based on learning. The experiments conducted on SVHN dataset and ICDAR, IIIT5K datasets show that STS-GAN has the ability to synthesize high-quality text images within a complex environment. In addition, compared to the non-learning methods, our STS-GAN can better fit the desired data distribution in natural scene images. Specifically, the main contributions of this paper include: (1) we construct STS-GAN to synthesize text in natural images, which is an entirely learning-based model using well-designed conditional GANs. The STS-GAN is designed as a novel two-stage architecture aiming to capture structural patterns in text images. (2) A novel style loss based on feature variance minimization is proposed to generate character images in harmonious patterns. (3) Experiments on challenging scene text datasets show STS-GAN is able to synthesize highly deceptive text images. In addition, the synthetic images can be used to boost the performance of a text recognition algorithm.

2 Background

The generative adversarial networks (GANs) aim to learn a mapping function G from random noise vector $\boldsymbol{z}$ to image $\boldsymbol{x}$ [Goodfellow *et al.*, 2014]. The mapping function $G : \boldsymbol{z} \rightarrow \boldsymbol{x}$, named generator, is adversarially trained with a discriminator D through the following objective function:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{\boldsymbol{x}'}[\log D(\boldsymbol{x}')] + \mathbb{E}_{\boldsymbol{z}}[\log(1 - D(G(\boldsymbol{z}))], \quad (1)$$

where $\boldsymbol{x}'$ is the real image. G tries to minimize this objective while D tries to maximize it, i.e., $G^* = \arg\min_G \max_D \mathcal{L}_{GAN}(G, D)$. Based on Equation 1, conditional GANs are proposed to control the generation by injecting supervised information y into both G and D. The formulas below are a hinge based method [Tran *et al.*, 2017] to train G and D iteratively:

$$\mathcal{L}_{cGAN}(D) = \mathbb{E}_{\boldsymbol{x}',y}[\max(0,\ 1 - D(\boldsymbol{x}', y))] + \mathbb{E}_{\boldsymbol{z},y}[\max(0,\ 1 + D(G(\boldsymbol{z}, y), y))], \quad (2)$$

$$\mathcal{L}_{cGAN}(G) = -\mathbb{E}_{\boldsymbol{z},y}[D(G(\boldsymbol{z}, y),\ y)], \quad (3)$$

where y typically could be labels in class-conditional image synthesis.

Showing promising performance in image synthesis, GANs have been further improved to impart stable training and high-quality images by many recent works. For example, to stabilize the training, Miyato *et al.* [2018] introduce spectral normalization (SN) that constrains the Lipschitz constant of the discriminator. Further, projection discriminator [Miyato and Koyama, 2018] is proposed taking an inner product between the embedded class vector and the feature vector, which improves class-conditional performance greatly. Other works attempt to explore helpful tricks for GANs, such as BigGANs [Brock *et al.*, 2019] which trains networks with larger scale (network channels and training batch), achieving impressive results on ImageNet.

The powerful capability makes GANs competent for many specific tasks. Ledig *et al.* [2017] train a GAN model combining with traditional content loss for image super-resolution. In [Isola *et al.*, 2017], an image-conditional model, called Pix2Pix, is proposed for image-to-image translation problems. Also based on conditional GANs, Reed *et al.* [2016] develop GAN-CLS to synthesize images given text descriptions. These methods above mainly focus on generating images with few structural scenes. MC-GAN [Azadi *et al.*,

2018] pays attention to designing glyphs, which first generates glyph then transfers color and ornamentation. Nevertheless, MC-GAN synthesizes each character images independently without sequence information. In [Liu *et al.*, 2018], canonical rendering process is applied to synthesize text images and the adversarial loss is introduced for feature learning, which is also deviated from our method that synthesizes text images from scratch.

3 Scene Text Synthesis GAN

Mathematically, we denote the characters collection as $\mathcal{C} = \{y_i\}_{i=1}^{N}$, where N is the number of characters. Given a text vector $\boldsymbol{y} \in C^M$, where M is text length, our goal is to generate corresponding image collection $\mathcal{X}$, that each image $\boldsymbol{x} \in \mathcal{X}$ is a readable representation of $\boldsymbol{y}$ in natural image. Figure 1 depicts the pipeline of our approach. To generate an image that contains characters sequence $\boldsymbol{y}$, we divide this procedure into two stages: character synthesis and word synthesis. In the following sections, we will detail our approach from the aspects of character synthesis and word synthesis.

3.1 Character Synthesis

Character synthesis stage describes the skeleton for a text image. Taking character label $y \in \mathcal{C}$ and latent vector $\boldsymbol{z}$ as input, this stage samples character image $\boldsymbol{x}_c$ by directly applying the conditional GAN. The scale of $\boldsymbol{x}_c$ is set to 32×32 pixels.

Variance-Minimization Style Loss

Even though the conditional GAN is able to generate legible characters as expected, it is impractical to integrate all the character images into text patch due to the high diversity of styles between each character (Figure 4a). Therefore, we try to explore a method to unify the style of each character in the same image. Within our framework, for a fixed latent vector $\boldsymbol{z}$, the output images of different $y \in C$ should visually be consistent in font, color, background, etc. However, the conditional GANs are powerless for this and the style of $\boldsymbol{x}_c \in \{G_c(\boldsymbol{z}, y)|y \in \mathcal{C}\}$ is out of control.

Suppose $\mathcal{S} \in \mathbb{R}^n$ is a style space that closer points in $\mathcal{S}$ have similar styles. Given a mapping function $\psi(\cdot)$ that projects $\boldsymbol{x}_c$ to corresponding representation in $\mathcal{S}$, we can apply $\psi(\cdot)$ to guide the training of G_c by restricting the distances of synthetic images in $\mathcal{S}$, so that G_c can generate character images with consistent style. For all the synthetic images $\boldsymbol{x}_c \in \{G_c(\boldsymbol{z}, y)|y \in \mathcal{C}\}$, minimizing the variance of corresponding representations in $\mathcal{S}$ means reducing the style discrepancy. Based on this observation, we propose the following objective:

$$\mathcal{L}_{style}(G_c) = \mathbb{E}_{\boldsymbol{z}}[Var_y[\psi(G_c(\boldsymbol{z}, y))]]. \tag{4}$$

The style function $\psi(\cdot)$ can be parametrized as a neural network. We share the network of $\psi(\cdot)$ with discriminator D_c, as it adds negligible computation to GAN training. Thus, Equation 4 can be rewritten as:

$$\mathcal{L}_{style}(G_c, D'_c) = \mathbb{E}_{\boldsymbol{z}}[Var_y[D'_c(G_c(\boldsymbol{z}, y))]]. \tag{5}$$

We note that Equation 5 can be easily approximated with Monte Carlo simulation. For the estimation of Var_y, we sample 8 labels for each latent vector $\boldsymbol{z}$. Then the mean value of each feature dimension in $\mathcal{S}$ is calculated after variance estimation. Therefore, Equation 5 can be optimized by traditional stochastic gradient descent together with GAN loss.

$\boldsymbol{z} \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$
fc, $4 \times 4 \times ch_1$
ResBlock up ch_2
ResBlock up ch_3
ResBlock up ch_4
BN, ReLU, 3×3 conv 3
Tanh

(a) Character generator

image $\boldsymbol{x} \in \mathbb{R}^{32 \times 32 \times 3}$
ResBlock down ch_4
ResBlock down ch_3
ResBlock ch_2
ResBlock ch_1, ReLU
Global sum pooling
Embed(y)·$\boldsymbol{h}$ + (fc $\rightarrow$ 1)

(b) Character discriminator

Table 1: Network architectures for character model. ch_i is the channel width of ResBlock.

In summary, combining conditional adversarial loss and style loss, our training objectives for D_c and G_c are concluded as follows:

$$\mathcal{L}_{D_c, D'_c} = \mathcal{L}_{cGAN}(D_c) + \lambda_s \mathcal{L}_{style}(D'_c), \tag{6}$$

$$\mathcal{L}_{G_c} = \mathcal{L}_{cGAN}(G_c) + \lambda_s \mathcal{L}_{style}(G_c), \tag{7}$$

where λ_s is a balance parameter between the GAN loss and style loss.

Our style loss is related to perceptual loss [Dosovitskiy and Brox, 2016; Gatys *et al.*, 2016] as both of us employ an additional network for feature space projection. However, our method differs from perceptual loss as we attempt to reduce feature variance between synthetic images from the same latent vector $\boldsymbol{z}$, rather than minimizing the norm of feature vectors between real and fake images.

Network Architecture

Our network architectures are designed mainly following [Miyato *et al.*, 2018; Miyato and Koyama, 2018], and the details are given in Table 1. Overall, we employ ResBlock defined in [Miyato and Koyama, 2018] as the backbone of both generator and discriminator. Spectral normalization is applied to each parameter layer both in generator and discriminator. For generator G_c, class label y is injected using class-conditional batch normalization [Dumoulin *et al.*, 2017]. For discriminator D_c, y is provided to D_c based on projection [Miyato and Koyama, 2018]. In addition, the style function in Equation 5 shares the network with discriminator D_c. We extract the output of global sum pooling in Table 1b as the style representation, so that $D'_c(G_c(\boldsymbol{z}, y)) \in \mathbb{R}^{ch_1}$, where ch_1 is the channel width of the last ResBlock in discriminator network.

3.2 Word Synthesis

By now we can synthesize character sequence in images by simply applying image stitching based on the fruit of Section 3.1. However, in this case there exist seams and discontinuities between characters, causing pool visual effects (top rows in Figure 6). In the word synthesis stage, we aim to fine-tune the stitched image $\hat{\boldsymbol{x}}_w$ to more visually appealing image $\boldsymbol{x}_w$. The stitched image $\hat{\boldsymbol{x}}_w$, whose aspect ratio is

	Encoder		Decoder
1	conv 64, SN, ReLU	6	deconv 512, SN, IN, ReLU
2	conv 128, SN, IN, ReLU	7	deconv 256, SN, IN, ReLU
3	conv 256, SN, IN, ReLU	8	deconv 128, SN, IN, ReLU
4	conv 512, SN, IN, ReLU	9	deconv 64, SN, IN, ReLU
5	conv 512, SN, IN, ReLU	10	deconv 3, SN, Tanh

Table 2: U-Net word generator.

kept invariant, i.e. the size is $32 \times (M \cdot 32)$, is fed into word model together with its character labels $\boldsymbol{y} \in \mathcal{C}^M$. We also design the word model following the idea of conditional GANs, in which the word generator is a U-Net [Ronneberger *et al.*, 2015] implementing pixel-wise manipulation.

Training Objective

To refine $\hat{\boldsymbol{x}}_w$ to $\boldsymbol{x}_w$, two aspects should be taken into account. The first is how to retain text structure in $\hat{\boldsymbol{x}}_w$. The second is how to remove noise and achieve realism, i.e., removing discontinuities and rearranging the characters layout. To preserve local appearance, we introduce $L1$ loss [Isola *et al.*, 2017; Shrivastava *et al.*, 2017] for self-regularization:

$$\mathcal{L}_{L1} = ||\boldsymbol{x}_w - \hat{\boldsymbol{x}}_w||_1. \tag{8}$$

To generate realistic images, we turn to a variant of conditional GAN. Different from character synthesis that applies character label y to Equation 2,3, we use word label $\boldsymbol{y}$ to control the content instead. Therefore, D_w and G_w can be optimized by minimizing the following objectives:

$$\begin{aligned}\mathcal{L}_{D_w} = &\mathbb{E}_{\boldsymbol{x}'_w,\boldsymbol{y}}[\max(0,\ 1 - D_w(\boldsymbol{x}'_w, \boldsymbol{y}))] + \\ &\mathbb{E}_{\boldsymbol{x}_w,\boldsymbol{y}}[\max(0,\ 1 + D_w(\boldsymbol{x}_w, \boldsymbol{y}))],\end{aligned} \tag{9}$$

$$\mathcal{L}_{G_w} = -\mathbb{E}_{\boldsymbol{x}_w,\boldsymbol{y}}[D_w(\boldsymbol{x}_w, \boldsymbol{y})] + \lambda_l \mathcal{L}_{L1}, \tag{10}$$

where $\boldsymbol{x}_w = G_w(\hat{\boldsymbol{x}}_w, \boldsymbol{y})$ and λ_l is a balance parameter between the GAN loss and $L1$ loss.

As GANs tend to disrupt the text structure obtained from $\hat{\boldsymbol{x}}_w$, the $L1$ self-regularization is crucial to preserve character appearance. Compared to generating text image from scratch, pixel-wise manipulation with self-regularization substantially reduces the underlying sample space, which eases the burden on GAN synthesis. Different from the $L1$ loss in [Isola *et al.*, 2017] applied between real and fake images, we restrict the output of the generator to its input. Our method also deviates from [Shrivastava *et al.*, 2017], as we explicitly consider word label $\boldsymbol{y}$ both in generator and discriminator.

Pixel-wise Manipulation and Patch Adversarial Learning

A basic demand of word model is to deal with variable-length images as the size of $\hat{\boldsymbol{x}}_w$ is $32 \times (M \cdot 32)$. Therefore, we implement both generator G_w and discriminator D_w as fully convolutional networks. For the generator G_w, we design the network architecture following U-Net, allowing G_w to manipulate images from pixels to pixels. Table 2 details the architecture, where conv and deconv are convolution and transposed convolution both with kernel 4×4 and stride 2; IN is instance normalization [Vedaldi, 2016] conditioned on $\boldsymbol{y}$. The feature maps from layer i and $n - i$ are concatenated, and then fed to layer $n - i + 1$, where $n = 10$ is the total number of layers.

As a strong discriminator is substantially important for providing correct gradient to generator training, we exploit several schemes to build D_w. First, D_w adopts the backbone of D_c (Table 1b) as its network, thus D_w can be initialized with well-trained D_c, which enables D_w to distinguish text patch initially. More concretely, we drop global sum pooling in Table 1b, resulting in an output response $\boldsymbol{r} \in \mathbb{R}^{8\times(M\cdot 8)}$, and the fully-connected (fc) layer is replaced by a 1×1 convolution layer. Note that fc layer can be viewed as 1×1 convolution operated on feature maps with shape 1×1, so that the 1×1 convolution layer in D_w can be initialized with fc layer in D_c. Then, we employ adversarial loss to local image patches [Li and Wand, 2016] rather than the full image, as it helps to keep local character structure. In our method, the local image patches are at the scale of 48×48 (if available), which is the size of receptive field in $\boldsymbol{r}$. We apply Equation 9 to each element in $\boldsymbol{r}$, and average the results to get the final discriminator response.

To inject labels into G_w and D_w, a straightforward way is assigning the label of each pixel in input image to its corresponding location in the feature map. More formally, $p_{i,j}$ is a pixel in fake image $\boldsymbol{x}_w$ or real image $\boldsymbol{x}'_w$, and s is the stride from image to feature map $\boldsymbol{f} \in \mathbb{R}^{C\times(32/s)\times(32\cdot W/s)}$, where C is the number of channels. Thus, $p'_{i/s,j/s}$ in $\boldsymbol{f}$ has the same label y with $p_{i,j}$. Besides, the labels in $\boldsymbol{x}_w$ and $\boldsymbol{x}'_w$ are easily obtained according to the image stitching and annotations, respectively. After determining the labels for feature maps, class-conditional instance normalization in generator and projection in discriminator can be conducted.

4 Experiments

4.1 Datasets and Evaluation Metrics

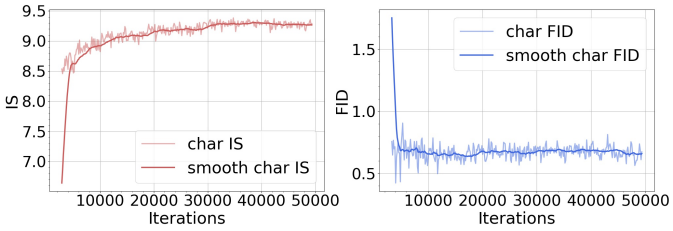
Datasets

We evaluate the proposed STS-GAN on two scene text datasets. The first is Street View House Number (SVHN) [Netzer *et al.*, 2011], which contains 10 classes from digit 1 to 10. There are 73257 training and 26032 test character images, and 33402 training and 13068 test word images in SVHN dataset. The second is an alphanumeric (62 characters) dataset composed of data from ICDAR 2003 [Lucas *et al.*, 2003] and IIIT 5K-word [Mishra *et al.*, 2012] datasets. This dataset (IC03+IIIT) has 15791 training and 20648 test character images. We keep the images with no more than twenty characters, thus the number of training and test word images are 3096 and 4056, respectively.

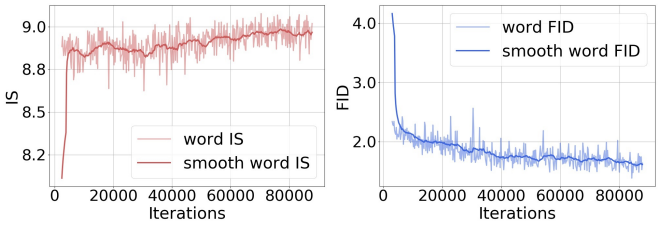
Evaluation Metrics

We primarily apply Inception score (IS) [Salimans *et al.*, 2016] to quantitatively evaluate the synthetic text images, as a high score generally indicates better visual appearance. In addition, Fréchet Inception distance (FID) [Heusel *et al.*, 2017] is used as an assistant tool to assess the realism and variation. Different from IS, lower FID means better. Rather than employing Inception network [Szegedy *et al.*, 2016] trained on ImageNet, we use ResNet classifiers [He *et al.*, 2016] trained on SVHN or IC03+IIIT datasets for IS and FID. Specifically,

Figure 2: Samples from SVHN dataset. The top rows are synthetic images, and the bottom row is the real images.



(a) Character model.



(b) Word model.

Figure 3: Training curves on SVHN for character and word models.

the ResNet classifiers are originally designed for CIFAR-10 with 56 layers. For character classifier, the size of input image is 32×32, and the feature vector before the last fully-connected layer is adopted to calculate the FID. For word classifier, the size of input image is 32×128. We share the backbone of character classifier with word classifier, which is equipped with an LSTM layer trained with CTC loss. The feature vector used for FID is the output of the last residual block. In all experiments, 50k samples are randomly sampled to compute the IS and FID scores for each evaluation.

Implementation Details

We employ Adam optimizer as solver with momentum $\beta_1 = 0$ and $\beta_2 = 0.999$ for all the networks. The learning rates are $2 \cdot 10^{-4}$ for D_c and $5 \cdot 10^{-5}$ for G_c, and $2 \cdot 10^{-4}$ for both D_w and G_w. For character model, batch size is set to 512. For word model, we sample 8 images with the same word length each batch. All the latent vector $\boldsymbol{z}$ is sampled from standard Gaussian distribution.

4.2 Evaluation on SVHN

To validate the effectiveness of STS-GAN, we first conduct experiments on a digit dataset. The margins of both character and word images are removed, thus the text is center-cropped in the images. Figure 2 presents some synthetic text images (top rows). Compared to the real images (bottom row), the fake images with arbitrary text length are highly deceptive.

Batch	ch_1, ch_2, ch_3, ch_4	Param (M)	IS	FID
512	256,128,64,32	1.879	39.14($\pm$0.29)	9.05
512	128,128,128,128	2.212	38.71($\pm$0.33)	6.98
512	512,256,128,64	6.252	39.70($\pm$0.37)	8.56
512	256,256,256,256	8.095	41.96($\pm$0.32)	9.06
512	1024,512,256,128	22.482	41.50($\pm$0.40)	9.46
128	256,256,256,256	8.095	44.06($\pm$0.33)	10.31
256	256,256,256,256	8.095	43.53($\pm$0.29)	9.78
1024	256,256,256,256	8.095	40.06($\pm$0.33)	8.38
2048	256,256,256,256	8.095	42.09($\pm$0.45)	8.91

Table 3: Evaluation of character model at different channel width and batch size. ch_i represents the number of channels in Table 1. Results are computed across 5 different random initializations.

Figure 3 plots the IS and FID training curves for character model (top) and word model (bottom). On one hand, the converged curves indicate STS-GAN is able to synthesize better images with increasing iterations. On the other hand, they also prove that by directly transferring IS and FID to our tasks, IS and FID can be effective metrics to evaluate the generation of text images, even though the character classifier and word classifier are newly trained on text image datasets.

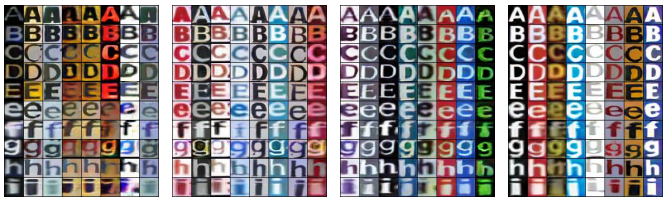
4.3 Evaluation on ICDAR and IIIT5K

To have a deep insight into STS-GAN, we further conduct experiments on a more challenging IC03+IIIT dataset. Compared to SVHN, this dataset has more character classes, more flexible styles, longer text and less training images.

Analysis of Character Synthesis

Network architecture. [Brock *et al.*, 2019] reports GANs benefit from wider network and larger batch size. We also conduct similar experiments (Table 3) on character model. To test the impact of network parameters, we gradually increase the number of channels (rows $1-5$). However, we do not notice wider networks obviously improve the performance, even the largest model (row 5) is slightly worse than a small one (row 4). Further, the batch size is doubled from 128 to 2048 (row $4, 6-9$) [1]. Also, this trick does not improve the performance simultaneously in IS and FID. We conjecture that the scaling trick explored in ImageNet suffers from saturation due to the limited training data in IC03+IIIT dataset. Therefore, we adopt the configuration in row 4 considering the trade-off between image quality and training efficiency.

[1] We implement batch size 1024, 2048 by accumulating gradients, thus the statistics of BatchNorm is computed at batch size 512.



(a) $\lambda_s = 0$, IS $= 42.96$, FID $= 10.13$ (b) $\lambda_s = 0.5$, IS $= 42.03$, FID $= 8.94$ (c) $\lambda_s = 1$, IS $= 42.16$, FID $= 9.23$ (d) $\lambda_s = 2$, IS $= 42.70$, FID $= 9.45$

Figure 4: Character samples from G_c trained with different λ_s. Each column uses the same latent vector $\boldsymbol{z}$.

$L1$	patch	pretrained	$D_w(\cdot, \boldsymbol{y})$	$G_w(\cdot, \boldsymbol{y})$	IS	FID
✗	✗	✗	✗	✗	5.8($\pm$0.1)	20.3
✓	✗	✗	✗	✗	7.5($\pm$0.1)	18.1
✓	✓	✗	✗	✗	26.3($\pm$0.6)	3.1
✓	✓	✓	✗	✗	28.5($\pm$0.6)	2.7
✓	✓	✓	✓	✗	31.5($\pm$0.6)	2.4
✓	✓	✓	✓	✓	32.4($\pm$0.8)	2.4

Table 4: Ablation study for word model. Results are computed across 3 different random initializations.

Style loss. Further, we investigate the effectiveness of proposed style loss by adjusting λ_s in Equation 6,7. As shown in Figure 4, λ_s is set to $0, 0.5, 1, 2$, respectively. Note that $\lambda_s = 0$ equals to training G_c without style loss. By comparing Figure 4a with 4c,4d, we can obviously observe that the style loss essentially unifies the font, color, background, etc. of the characters. However, a small λ_s (Figure 4b) is too weak to control the style. Though a big λ_s (Figure 4d) achieves satisfactory style appearance, we find it slowing down the convergence slightly. Thus, $\lambda_s = 1$ is adopted in our further experiments. Besides, the style loss does not harm performance according the IS and FID scores.

Analysis of Word Synthesis

Ablation study. Applying vanilla GAN with U-Net as the generator and removing the latent vector, we build a simple baseline. Based on this baseline, we illustrate the effectiveness of each component of the proposed method. Table 4 summarizes the comparison results, where $L1$ is the loss defined in Equation 8 ($\lambda_l = 1$); *patch* represents the patch discriminator; *pretrained* indicates initializing D_w using D_c; $D_w(\cdot, \boldsymbol{y})$ and $G_w(\cdot, \boldsymbol{y})$ are imposing condition $\boldsymbol{y}$ on discriminator and generator respectively. As can be seen from Table 4, without $L1$ and *patch* (rows 1,2), the models fail to generate plausible images. The configuration of 3rd row is quite similar to [Shrivastava *et al.*, 2017] as *L1* and *patch* are the principal components of [Shrivastava *et al.*, 2017]. Further, we gain benefit from using pretrained D_w (4th row). Lastly, injecting $\boldsymbol{y}$ to D_w substantially improves the IS and FID scores, and G_w conditioned on $\boldsymbol{y}$ also boosts the IS score.

L1 loss. Based on the results of ablation study, we further analyze the impact of $L1$ self-regularization, which is important to retain character appearance from the character model. As reported in Table 5, a big λ_l typically reduces the image diversity and a small one is hard to keep the character skeleton. Therefore, we choose $\lambda_l = 5$ as it performs the best performance both in realism and variation.

λ_l	0.1	0.2	0.5	1	2	5	10	20
IS	26.6 ($\pm$0.6)	29.1 ($\pm$0.6)	31.1 ($\pm$0.4)	32.4 ($\pm$0.8)	32.8 ($\pm$0.5)	33.3 ($\pm$0.5)	32.9 ($\pm$0.5)	32.3 ($\pm$0.7)
FID	2.9	2.6	2.5	2.4	2.5	2.5	2.7	3.1

Table 5: Impact of $L1$ coefficient. Results are computed across 3 different random initializations.

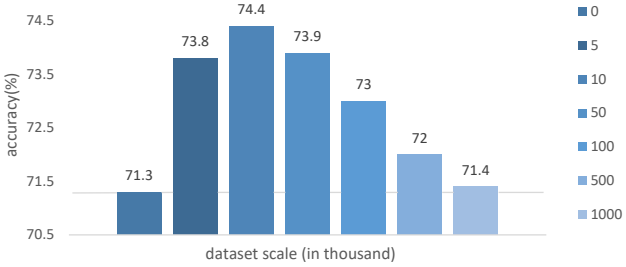


Figure 5: Recognition accuracy of word classifiers trained with IC03+IIIT datasets and additionally nk synthetic images.

Scene text recognition. Finally, we apply the proposed STS-GAN to scene text recognition [Fang *et al.*, 2018; Xie *et al.*, 2019]. To obtain high-quality images, we only keep the synthetic images which are classified correctly by the word classifier with each character confidence greater than 0.9. Therefore, we build a synthetic dataset, used for training the word classifier together with IC03+IIIT dataset. From the accuracy comparisons in Figure 5 we can see, training with additional synthetic images can boost the performance for word classifiers (up to 3.1% for 10k), which also demonstrates the verisimilitude of synthetic images as unreadable images will result in turbulent training. However, more synthetic images do not help to improve accuracy linearly. We speculate training with limited data, the image diversity of our method is unable to support the accuracy growth in large scale, which remains further study. In total, starting from only 3096 training word images, our STS-GAN is able to synthesize readable images and boost the recognition performance for a mainstream algorithm.

4.4 Comparison with Other Models

Finally, we compare the performance of the proposed model with non-learning based methods and traditional GAN mod-

Type	Method	IS	FID
non-learning	Jaderberg *et al.* [2014]	29.4($\pm$0.4)	4.5
	Gupta *et al.* [2016]	28.1($\pm$0.6)	3.8
	Zhan *et al.* [2018]	30.1($\pm$0.6)	3.7
learning	GAN-CLS [Reed *et al.*, 2016]	7.2($\pm$0.3)	17.5
	Character model + Pix2Pix [Isola *et al.*, 2017]	8.6($\pm$0.4)	15.4
	STS-GAN	**32.4**($\pm$**0.8**)	**2.4**

Table 6: Comparison with other methods using IS and FID metrics.

Figure 6: Synthetic samples from IC03 and IIIT datasets. The top rows are stitched images, and the bottom rows are corresponding synthetic images.

(a) GAN-CLS

(b) Character model + Pix2Pix

Figure 7: Samples from traditional GAN models.

Figure 8: Example failure cases. The odd rows are stitched images, and the even rows are synthetic images.

els, and the IS and FID scores are recorded in Table 6. The non-learning methods [Jaderberg *et al.*, 2014; Gupta *et al.*, 2016; Zhan *et al.*, 2018] achieve impressive IS and FID scores as they are effective to generate readable text images. Note that our method synthesizes text images from scratch without complicated rendering process. Compared to the non-learning methods, our STS-GAN obtains better IS and FID scores, which denotes that STS-GAN not only generates high-quality images, but also captures an approximately correct distribution as the desired image distribution (i.e., IC03+IIIT dataset in this case).

In addition, two learning-based methods are prepared to adapt our task. The first is GAN-CLS [Reed *et al.*, 2016], which is a conditional GAN designed for description text to image synthesis. Sentence embedding in GAN-CLS is replaced with the embedding of text label $\boldsymbol{y}$ to make it support our task, and $\boldsymbol{y}$ is embedded using ELMo [Peters *et al.*, 2018]. The second is a two-stage method using Pix2Pix [Isola *et al.*, 2017] to replace our word synthesis. Figure 7 displays some samples and Table 6 gives corresponding IS, FID scores for the above two models. The GAN-CLS (Figure 7a) merely generates confusing and meaningless images, as vanilla GANs are hard to recover character structure. Also, for the popular image-to-image framework (Figure 7b), it is difficult to retain clear text appearance obtained from the character model. Compared to these models, our well-designed STS-GAN can successfully synthesize images with clear and readable text.

4.5 Discussion

Drawing text in natural images is a challenging problem. Though STS-GAN is demonstrated to be effective, we still observe some failure cases in IC03+IIIT dataset, which rarely occur in easier SVHN dataset. Generally, with a stitched image $\hat{\boldsymbol{x}}_w$ in which text is legible comes a well-drawn $\boldsymbol{x}_w$. In some cases, we have noticed successful images converted from poor $\hat{\boldsymbol{x}}_w$ (Figure 6). Nevertheless, failure conversion will cause a clear $\hat{\boldsymbol{x}}_w$ to an unsatisfactory $\boldsymbol{x}_w$, and a blurring $\hat{\boldsymbol{x}}_w$ typically leads to failure synthesis (Figure 8). Besides, we note that in character synthesis stage, due to the limited training samples, classes with low character frequency ('Q', 'q', 'j', etc.) encounter mode collapse, which may also harm the final performance.

5 Conclusion

In this paper, we propose scene text synthesis GAN (STS-GAN) for generating text images. To cope with the problem that vanilla GANs are difficult to model structural patterns, we first generate local character structure which is constrained by style loss to unify text style. Then local characters are converted into plausible word image through pixel-wise manipulation. We conduct experiments on SVHN and ICDAR, IIIT5K datasets, showing STS-GAN can synthesize visually pleasing text images. In the future, we will extend this method to full scene text images rather than cropped images.

Acknowledgments

This work is supported by National Key R&D Program 2016 (Grant No. 2016YFB0801305), National Defense Science and Technology Fund for Distinguished Young Scholars (2017-JCJQ-ZQ-022), the National Nature Science Foundation of China (61525206, 61771468), the Youth Innovation Promotion Association Chinese Academy of Sciences (2017209).

References

[Azadi *et al.*, 2018] Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer. In *CVPR*, pages 7564–7573, 2018.

[Brock *et al.*, 2019] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019.

[Dosovitskiy and Brox, 2016] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, 2016.

[Dumoulin *et al.*, 2017] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *ICLR*, 2017.

[Fang *et al.*, 2018] Shancheng Fang, Hongtao Xie, Zheng-Jun Zha, Nannan Sun, Jianlong Tan, and Yongdong Zhang. Attention and language ensemble for scene text recognition with convolutional sequence modeling. In *ACM MM*, pages 248–256, 2018.

[Gatys *et al.*, 2016] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[Gupta *et al.*, 2016] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Heusel *et al.*, 2017] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

[Isola *et al.*, 2017] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[Jaderberg *et al.*, 2014] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *NIPS Deep Learning Workshop*, 2014.

[Ledig *et al.*, 2017] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Cunningham, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690, 2017.

[Li and Wand, 2016] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, pages 702–716, 2016.

[Liu *et al.*, 2018] Yang Liu, Zhaowen Wang, Hailin Jin, and Ian Wassell. Synthetically supervised feature learning for scene text recognition. In *ECCV*, pages 435–451, 2018.

[Lucas *et al.*, 2003] Simon M Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, and Robert Young. Icdar 2003 robust reading competitions. In *ICDAR*, 2003.

[Mishra *et al.*, 2012] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012.

[Miyato and Koyama, 2018] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *ICLR*, 2018.

[Miyato *et al.*, 2018] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, et al. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop*, volume 2011, page 5, 2011.

[Peters *et al.*, 2018] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[Reed *et al.*, 2016] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, pages 1060–1069, 2016.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[Salimans *et al.*, 2016] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016.

[Shrivastava *et al.*, 2017] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017.

[Szegedy *et al.*, 2016] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[Tran *et al.*, 2017] Dustin Tran, Rajesh Ranganath, and David M. Blei. Hierarchical implicit models and likelihood-free variational inference. In *NIPS*, 2017.

[Vedaldi, 2016] Victor Lempitsky Dmitry Ulyanov Andrea Vedaldi. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint:1607.08022*, 2016.

[Xie *et al.*, 2019] Hongtao Xie, Shancheng Fang, Zheng-Jun Zha, Yating Yang, Yan Li, et al. Convolutional attention networks for scene text recognition. *ACM TOMM*, 2019.

[Zhan *et al.*, 2018] Fangneng Zhan, Shijian Lu, and Chuhui Xue. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In *ECCV*, 2018.

[Zhang *et al.*, 2018] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint:1805.08318*, 2018.