Synthetically Supervised Feature Learning for Scene Text Recognition

Yang Liu[1(✉)], Zhaowen Wang[2], Hailin Jin[2], and Ian Wassell[1]

[1] Computer Laboratory, University of Cambridge, Cambridge, UK
{yl504,ijw24}@cam.ac.uk
[2] Adobe Research, San Jose, CA, USA
{zhawang,hljin}@adobe.com

Abstract. We address the problem of image feature learning for scene text recognition. The image features in the state-of-the-art methods are learned from large-scale synthetic image datasets. However, most methods only rely on outputs of the synthetic data generation process, namely realistically looking images, and completely ignore the rest of the process. We propose to leverage the parameters that lead to the output images to improve image feature learning. Specifically, for every image out of the data generation process, we obtain the associated parameters and render another "clean" image that is free of select distortion factors that are applied to the output image. Because of the absence of distortion factors, the clean image tends to be easier to recognize than the original image which can serve as supervision. We design a multi-task network with an encoder-discriminator-generator architecture to guide the feature of the original image toward that of the clean image. The experiments show that our method significantly outperforms the state-of-the-art methods on standard scene text recognition benchmarks in the lexicon-free category. Furthermore, we show that without explicit handling, our method works on challenging cases where input images contain severe geometric distortion, such as text on a curved path.

Keywords: Scene text recognition · Deep learning · Neural networks
Feature learning · Synthetic data · Multi-task learning

1 Introduction

Scene text recognition, the problem of recognizing text in natural scene images, has always occupied a special place in image understanding and Computer Vision, because of the importance of text in the way that people communicate with each other. It has a wide range of practical applications including autonomous driving, robots and drones, mobile e-commerce, and helping visually impaired people. Image features play a crucial role in scene text recognition. Early methods use hand-crafted features and break the problem into subproblems such as character detection [34,36,38]. The state-of-the-art methods use convolutional neural networks and train from images directly to text in an end-to-end fashion [17,27].

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11209, pp. 449–465, 2018.
https://doi.org/10.1007/978-3-030-01228-1_27

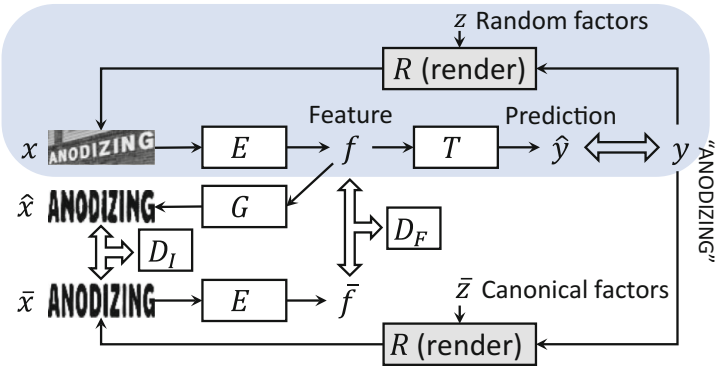


Fig. 1. The proposed text feature learning framework. The blue shaded box at the top contains a generic text recognition pipeline, with an input image $\mathbf{x}$ going through a feature encoder E and a text decoder T, resulting in a predicted text string $\hat{y}$. By a synthetically-supervised approach, we use the true text label y to render not only a noisy input image $\mathbf{x}$, but also a clean image $\bar{\mathbf{x}}$ with the canonical rendering parameter $\bar{\mathbf{z}}$. The encoded feature $\mathbf{f} = E(\mathbf{x})$ is trained to match its clean counterpart $\bar{\mathbf{f}}{=}E(\bar{\mathbf{x}})$, as well as to reproduce the clean image through an image generator G. Adversarial matching losses are imposed on both image and feature domains by discriminators D_I and D_F. (Color figure online)

One of the key factors in the state-of-the-art methods is the use of large-scale synthetic image datasets to train convolutional neural networks [17]. The ability to use synthetic data is special in the text recognition problem. Thanks to the fact that text is *not* a natural object, we are able to generate an unlimited amount of labeled images that resemble real-world images. In the generation process, we can manipulate nuisance factors such as font, lighting, shadow, border, background, image noise, geometric deformation, and compression artifacts. As a result, image features trained on synthetic data with these factors will be robust to their variations, leading to a significant improvement of recognition accuracy.

There is a fundamental difference between real images and synthetic images which is that synthetic images are obtained through a process that is controllable to a Machine Learning algorithm. This process provides not only an unlimited amount of training data (images and labels) but also parameters that are associated with the data. This difference has been completely ignored in the literature. For instance, most state-of-the-art methods follow a simple training procedure and only exploit the abundance of synthetic data to train image features. The key idea of this work is that we can leverage the difference between real and synthetic images, namely the controllability of the generation process, and control the generation process to generate paired training data. Specifically, for every synthetic image out of the generation process with aforementioned nuisance factors, we obtain the associated rendering parameters, manipulate the parameters, and generate a corresponding *clean* image where we remove part or all of the nuisance factors. For instance, the original image may have a perspective warp

and the clean image does not contain any geometric deformation. Because of the absence of nuisance factors, the text in the clean image is generally easier to recognize and can therefore serve as supervision. By training on synthetic images both with and without nuisance factors, we expect to learn a more robust text recognition feature that is invariant to undesired nuisance factors.

The overall framework of our proposed method, which we call synthetically supervised feature learning, is shown in the Fig. 1. We use cleans image as supervision at both the pixel level and the feature level in a generative way, and design auxiliary training losses that can be combined with conventional training objectives of any deep network model for text recognition. We follow two principles – *invariance* and *completeness* – to learn a good text feature encoder $E(\cdot)$, which usually consists of the first several convolutional layers in the recognition model. Feature invariance requires that the encoder extracts the same feature for any input image $\mathbf{x}$ and its corresponding clean image $\bar{\mathbf{x}}$: $E(\mathbf{x}) = E(\bar{\mathbf{x}})$. Feature completeness requires all text label information to be contained in $E(\mathbf{x})$. It is equivalent to require the existence of an inverse mapping, or an image generator $G(\cdot)$, that can transform the encoded feature back to the deterministic clean image: $G(E(\mathbf{x})) = \bar{\mathbf{x}}$. Since the supervision from the clean image is applied on image and feature domains, it is tempting to employ generative adversarial networks (GANs) [7] to help the feature learning in addition to the use of basic ℓ_1 or ℓ_2 losses. Therefore, we also explore using discriminators $D_I(\cdot)$ and $D_F(\cdot)$ to encourage the generated image and feature to be more similar to their clean counterparts, respectively. Our experiment results show that, with the right combination, the invariance, completeness and adversarial losses all contribute to a text feature that is more robust to nuisance factors.

The main contributions of this paper are threefold: 1. We propose to leverage the controllability of the data generation process and introduce clean images that are free byproducts as the auxiliary training data for scene text recognition. Otherwise, our method does not require information of other nuisance factors in the generation process which is less structured and harder to use. We propose a general algorithm to use clean images as additional supervision that can be applied to most deep learning based text recognition models. 2. We design a novel scene text recognition algorithm that learns a descriptive and robust text representation (image feature) through image generation, feature matching and adversarial training. We conduct a detailed ablation study by examining the effectiveness of each proposed component. 3. Our method achieves the state-of-the-art performance on various scene text recognition benchmarks and significantly outperforms the state-of-the-art in the lexicon-free category. Moreover, Our approach generalizes to irregular text recognition, such as perspective text and curved text recognition.

2 Related Work

Scene text recognition is an important area in image understanding and Computer Vision. There is a sizable body of literature on this topic. We will only discuss closely related work here and refer the reader to recent surveys [34,36,38]

for more thorough expositions. [14,15,32] are among the early works in using deep convolutional neural networks as image features for scene text recognition. [17] formulates the problem as a 90K-class convolutional neural network, where each class corresponds to an English word. One of the key contributions of [17] is that it proposes a large-scale synthetic dataset as existing image datasets are not sufficient to train deep convolutional neural networks. This synthetic dataset is later adopted by follow-up works. To overcome the problem of using a fixed lexicon in training, [16] proposes a joint graphical model and [27] propose an end-to-end sequence recognition network where images and texts are separately encoded as patch sequences and character sequences. A lexicon can be introduced at the test time if necessary. [4,5,20] are among the latest approaches which adopt attention-based networks to handle complicated text distortion and low-quality images. Our method follows the general direction of using convolutional neural networks and sequence recognition for the problem. Our contribution lies in using the rendering parameters in the synthetic data generation process to obtain new clean reference images. We leverage both original images and clean images to guide image feature learning. To the best of our knowledge, this is the first work in scene text recognition to use auxiliary reference images to improve feature learning, sharing similar philosophy with other generative multi-task learning works [24,30,35]. We show that our method can correct geometric distortion present in input images. This is related to [28] which uses a spatial transformer network to rectify the image before the recognition pipeline. However, [28] employs a hand-designed architecture that only works for geometric distortion while our method applies to arbitrary distortion in a unified way. As long as the synthetic data generation process can simulate a distortion, our method can potentially correct it through feature learning.

3 Method

We build a synthetically-supervised feature learning framework for text recognition as shown in Fig. 1. It consists of a text image renderer R, a feature encoder E, a text decoder T, an image generator G, and two discriminators D_I and D_F. We discuss each of these components and their interactions in the following.

Renderer: We use a standard text renderer R to synthesize a text image $\mathbf{x}{=}R(y, \mathbf{z})$ with a text string y and rendering parameters $\mathbf{z}$. $\mathbf{z}$ describes how nuisance factors are added in the rendered image, and is drawn randomly from a distribution covering the combinations of various factors including font, outline, color, shading, background, perspective warping, and imaging noise. The clean image $\bar{\mathbf{x}}$ for text y is synthesized as $R(y, \bar{\mathbf{z}})$ by fixing the rendering parameters to a canonical value $\bar{\mathbf{z}}$. In our case, $\bar{\mathbf{z}}$ corresponds to a standard font and zero noise perturbation, yielding a clean image $\bar{\mathbf{x}}$ as illustrated in Fig. 1. The renderer provides training triplets $\{(\mathbf{x}, \bar{\mathbf{x}}, y)\}$ in our framework, and it is not trainable.

Encoder and Text Decoder: The encoder E takes an input image $\mathbf{x}$ to extract its image feature $\mathbf{f}$, which is further fed into the text decoder T to predict the text character sequence $\hat{y}$. The cross-modal encoder-decoder structure represents a generic deep network design for scene text recognition. We follow the prior work of [27] to build these two components.

Specifically, E is a multi-layer fully convolutional network that extracts a 3D feature map $\mathbf{f}$, and T is a two-layer Bidirectional Long-Short Term Memory (BLSTM) network [10,11] that predicts text by solving a sequence labeling problem. The feature map $\mathbf{f}$ is first transformed to a sequence $\{\mathbf{f}^1, ..., \mathbf{f}^N\}$ by flattening N feature segments sliced from $\mathbf{f}$ horizontally from left to right. Due to the translation-invariant property of CNN, each feature frame $\mathbf{f}^n$ corresponds to the n-th local image region which may contain one or part of a text glyph. With the feature sequence as input, the BLSTM decoder T analyzes the dependency among the feature frames and predicts a character probability distribution π^n corresponding to each $\mathbf{f}^n$. The probability space of π^n includes all English alphanumeric characters as well as a blank token for word separation. Finally, the per-frame predictions $\{\pi^1, ..., \pi^T\}$ are translated into the text prediction $\hat{y}$ through beam search.

As in [27], the network branch of E and T can be trained by minimizing the discrepancy between the probability sequence $\{\pi^1, ..., \pi^T\}$ and the true text y using the Connectionist Temporal Classification (CTC) technique [9]. CTC aligns the variable length character sequence of y with the fixed length probability sequence so that the conditional probability of y can be evaluated based on $\{\pi^1, ..., \pi^T\}$. The training loss given by the direct supervision from y can be summarized as

$$\min_{E,T} \mathcal{L}_y = p(y|T(E(\mathbf{x}))) = \sum_{\tilde{y}:\mathcal{B}(\tilde{y})=y} \prod_{t=1}^{T} \pi^t(\tilde{y}^t), \tag{1}$$

where $\mathcal{B}$ is the CTC mapping for sequences of length T, and $\tilde{y}^t$ denotes the t-th token in $\tilde{y}$.

Feature Matching and Image Generator: Our motivation of utilizing the clean image $\bar{\mathbf{x}}$ is to learn a good text feature encoder E that is both invariant to nuisance factors and complete in describing text content. In terms of invariance, we explicitly minimize the difference between the features extracted from $\mathbf{x}$ and $\bar{\mathbf{x}}$, since the two images share the same text label y:

$$\min_{E} \mathcal{L}_f = \|E(\mathbf{x}) - E(\bar{\mathbf{x}})\|_2 . \tag{2}$$

In terms of completeness, we require all information in the clean image $\bar{\mathbf{x}}$ to be captured by feature $E(\mathbf{x})$. Equivalently, there should exist an image generator G that can reconstruct $\bar{\mathbf{x}}$ given $E(\mathbf{x})$. To generate images, we construct G as a deconvolutional network, which is trained jointly with the encoder E to minimize the ℓ_1 image reconstruction loss:

$$\min_{E,G} \mathcal{L}_g = \|G(E(\mathbf{x})) - \bar{\mathbf{x}}\|_1 . \tag{3}$$

Adversarial Discriminators: As the supervision from the clean image $\bar{\mathbf{x}}$ is applied on image and feature domains, we thus also explore the idea of generative adversarial network (GAN) [7] to help improve the distributional similarity between $G(E(\mathbf{x}))/E(\mathbf{x})$ and their clean counterparts $\bar{\mathbf{x}}/E(\bar{\mathbf{x}})$. We design an image discriminator D_I and a feature discriminator D_F that try to distinguish between noise and clean input sources. The two discriminators are both convolutional networks with binary classification outputs, and they are trained against E and G in an adversarial minimax style:

$$\min_{E,G}\max_{D_I}\mathcal{L}_{ga} = \log D_I(\bar{\mathbf{x}}|\mathbf{x}) + \log(1 - D_I(G(E(\mathbf{x}))|\mathbf{x})), \tag{4}$$

$$\min_{E}\max_{D_F}\mathcal{L}_{fa} = \log D_F(E(\bar{\mathbf{x}})) + \log(1 - D_F(E(\mathbf{x}))). \tag{5}$$

Note that the image discriminator D_I in Eq. (4) is formulated as a conditional GAN [22] conditioned on the original input image $\mathbf{x}$. This encourages the image generated by G to not only look realistic but also have the same text content as $\mathbf{x}$.

With all the above loss terms combined together, we come to the overall training objective for our synthetically-supervised text recognition model:

$$\min_{E,T,G}\max_{D_I,D_F}\mathbb{E}_{\mathbf{x},\bar{\mathbf{x}},y}\left[\mathcal{L}(\mathbf{x},\bar{\mathbf{x}},y)\right], \quad \mathcal{L} = \lambda_y\mathcal{L}_y + \lambda_f\mathcal{L}_f + \lambda_g\mathcal{L}_g + \lambda_{ga}\mathcal{L}_{ga} + \lambda_{fa}\mathcal{L}_{fa}, \tag{6}$$

where all the λ's are weighting coefficients. The effect of each individual loss and their best combinations will be discussed in the experiments.

4 Experiments

In this section, we evaluate our model on a number of benchmarks for scene text recognition. The network structure and implementation details are provided in Sect. 4.1. We present an ablation study in Sect. 4.2 to explore how the performance of the proposed method is affected by different model configurations, including different types of clean image $\bar{\mathbf{x}}$ and different combinations of model components. A comprehensive comparison on general recognition benchmarks is reported in Sect. 4.3. Finally, to further demonstrate the generalization capability of our proposed model, we verify its robustness on two benchmarks created especially for irregular text recognition in Sect. 4.4.

4.1 Implementation Details

Network Structure: Detailed information of the network structure is provided in Table 1. For the design of encoder E and text decoder T, we follow the configuration in [27] to enable a fair comparison. The BLSTM has 256 memory blocks and 37 output units (26 letters, 10 digits and 1 EOS symbol). The batch-normalization is applied after the 5^{th} and 6^{th} convolutional layers. Since the stability of the adversarial training suffers if sparse gradient layers are used,

Table 1. Network structure for our scene text recognition algorithm

Layer	Filter/stride	Output size	Layer	Filter/stride	Output size
Encoder			Image generator		
Input	-	$32 \times 100 \times 3$			
Conv1	$3 \times 3/2 \times 2$	$16 \times 50 \times 64$	FConv7	$2 \times 2/2 \times 1$	$2 \times 25 \times 512$
Conv2	$3 \times 3/2 \times 2$	$8 \times 25 \times 128$	FConv6	$3 \times 3/2 \times 1$	$4 \times 25 \times 512$
Conv3	$3 \times 3/1 \times 1$	$8 \times 25 \times 256$	FConv5	$3 \times 3/1 \times 1$	$4 \times 25 \times 256$
Conv4	$3 \times 3/2 \times 1$	$4 \times 25 \times 256$	FConv4	$3 \times 3/2 \times 1$	$8 \times 25 \times 256$
Conv5	$3 \times 3/1 \times 1$	$4 \times 25 \times 512$	FConv3	$3 \times 3/1 \times 1$	$8 \times 25 \times 256$
Conv6	$3 \times 3/2 \times 1$	$2 \times 25 \times 512$	FConv2	$3 \times 3/2 \times 2$	$16 \times 50 \times 128$
Conv7	$2 \times 2/2 \times 1$	$1 \times 25 \times 512$	FConv1	$3 \times 3/2 \times 2$	$32 \times 100 \times 3$
Feature discriminator			Image discriminator		
ConvF1	$1 \times 1/1 \times 1$	$1 \times 25 \times 256$	ConvI1	$3 \times 3/2 \times 2$	$16 \times 50 \times 64$
ConvF2	$1 \times 1/1 \times 1$	$1 \times 25 \times 128$	ConvI2	$3 \times 3/2 \times 2$	$8 \times 25 \times 128$
ConvF3	$1 \times 1/1 \times 1$	$1 \times 25 \times 64$	ConvI3	$3 \times 3/2 \times 1$	$4 \times 25 \times 256$
ConvF4	$1 \times 1/1 \times 1$	$1 \times 25 \times 32$	ConvI4	$3 \times 3/2 \times 1$	$2 \times 25 \times 256$
ConvF5	$1 \times 1/1 \times 1$	$1 \times 25 \times 1$	ConvI5	$2 \times 2/2 \times 1$	$1 \times 25 \times 1$
AvgPool	$1 \times 25/1 \times 1$	$1 \times 1 \times 1$	AvgPool	$1 \times 25/1 \times 1$	$1 \times 1 \times 1$
Text decoder					
BLSTM1	256	25×512			
BLSTM2	256	25×512			
Output	37	25×37			

Fig. 2. Example of different formations of clean images.

we replace MaxPool and ReLu with stride convolution and leaky rectified linear unit respectively. The image generator G contains a series of fractional-stride convolutions [2] to generate an image with the same size of the original input. The discriminators D_I and D_F both contain five fully convolutional layers.

Training Details: For all the experiments for scene text recognition, we use the synthetic dataset (Synth90) released by Jaderberg et al. [14] as the training data. The dataset contains 8 million images and their corresponding ground truth text labels. Different types of clean images are leveraged to supervise feature learning, and their effectiveness is analyzed in Sect. 4.2. Our network is trained on Synth90 and tested on all other real-world test datasets without any fine-tuning. Detailed information about real-world test benchmarks is provided in

Sects. 4.3 and 4.4. Following [27], images are resized to 32×100 in both training and testing. The image intensities are linearly scaled to the range of $[-1, 1]$. The batch size is set to 32. All weights are initialized from a zero-mean normal distribution with a standard deviation of 0.01. The Adam optimizer [19] is used with a learning rate of 0.002 and momentum 0.5. The parameters in the objective function (6) are determined by 5-fold cross-validation. For testing, in the process of unconstrained text recognition (lexicon-free), we straightforwardly select the most probable character. While in constrained text recognition, we calculate the conditional probability distributions for all lexicon words, and take the one with the highest probability as output result.

4.2 Ablation Study

In this section, we empirically investigate how the performance of the proposed method is affected by different model settings on the Street View Text dataset [31]. We study mainly in two aspects: the formation of clean image and the contribution of network components.

Formation of Clean Images: One of the main contributions of this paper is that we explore using clean image as auxiliary supervision to guide feature learning. To enable a fair comparison with existing works, our training data are the pre-rendered images from Synth90 [14], with the text labels being the only accessible rendering parameter. To evaluate the effects of removing different nuisance factors, besides rendering a clean image without any noise perturbation, we post-process the original input images to simulate the formation of different types of "less clean" images, as shown in Fig. 2, in the following ways.

Binarized Images: To remove image color variation, we convert an input image to gray-scale and then binarize the gray-scale image by thresholding. The threshold is set to be the mean value of the input image. The output binary image has 0 (black) for all pixels with intensity less than the mean value and 255 (white) otherwise.

Deskewed Images: To remove text orientation variation, we first detect the text baseline in the input image using a pre-trained neural network model for text detection [37]. Then we compute the angle of the text and rotate the text to the horizontal orientation.

Ideal Images: We render a new image which matches the ground truth text label while removing all the other nuisance factors. More specifically, we use the FreeType library [12] to render the corresponding text in black with font style 'Brevia Black Regular'. The font size is set as 64. The text is arranged horizontally in a clean white background. After rendering, we re-scale the synthesized image to 32×100, which has the same size as the original input image.

Table 2. Text recognition accuracy on SVT [31] using different types of clean images.

Clean image	Recognition accuracy (%)
None [27]	80.8
Binarized images	85.8
Deskewed images	84.7
Ideal images	**87.0**

Table 3. Text recognition accuracies for different variants of our model, compared with CRNN [27] baseline. The corresponding training losses are shown.

Model variant	Training losses	Accuracy (%)
CRNN [27]	$\mathcal{L}_y$	80.8
Image generation	$\mathcal{L}_y + \mathcal{L}_g$	86.1
Adversarial generation	$\mathcal{L}_y + \mathcal{L}_g + \mathcal{L}_{ga}$	84.7
Feature matching	$\mathcal{L}_y + \mathcal{L}_f$	85.1
Adversarial matching	$\mathcal{L}_y + \mathcal{L}_g + \mathcal{L}_f + \mathcal{L}_{fa}$	**87.0**

The performances of our model using 3 types of clean images are shown in Table 2, together with the CRNN model [27] trained without using any auxiliary clean data as a baseline. To enable a fair comparison, we use the same model architecture for all the clean image variants, and the configurations of our encoder and text decoder match those used in [27]. As shown in Table 2, introducing auxiliary clean data boosts the performance significantly. The reason is that removing part or all the nuisance factors from the original image makes text recognition easier. We further observe that leveraging the ideal image leads to the highest accuracy, which outperforms the baseline by over 6%. We attribute this improvement to that the ideal image makes the learned feature resilient to all the nuisance factors. The learned feature is optimized with respect to the text information while being invariant to other undesired nuisance factors, which is critical for scene text recognition. We use the ideal image as auxiliary supervision throughout the rest of the experiments.

Architectural Variants: We conduct a detailed ablation study by examining the effectiveness of each proposed component in our network structure. We evaluate and compare each of the following module configurations:

CRNN Model [27]**:** built with components E and T, and trained only with a CTC loss, corresponding to $\mathcal{L}_y$ in our framework.

Image Generation: built with E, T, and G, and trained with $\mathcal{L}_y$ and $\mathcal{L}_g$ losses.

Adversarial Generation: built with E, T, G and D_I, and trained with $\mathcal{L}_y$, $\mathcal{L}_g$ and $\mathcal{L}_{ga}$. Previous approaches have found it to be beneficial to mix the GAN objective with the ℓ_1 loss [13]. The encoder and the image generator work cooperatively to compete with the image discriminator.

Feature Matching: built with E and T, and trained with $\mathcal{L}_y$ and $\mathcal{L}_f$.

Adversarial Matching: built with E, T, G and D_F, and trained with $\mathcal{L}_y$, $\mathcal{L}_g$, $\mathcal{L}_f$ and $\mathcal{L}_{fa}$. The encoder not only tries to make the features of the original input and its corresponding clean image pair similar, but also to fool the feature discriminator. The adversarial game is conducted between the encoder and the feature discriminator. We also impose ℓ_1 reconstruction loss at pixel level.

The performances of the above 5 models are listed in Table 3. The CRNN model [27] serves as a baseline in the comparison. The 4 different variants of the proposed model all boost recognition performance compared to the baseline. Adding either feature consistency loss $\mathcal{L}_f$ or image generation loss $\mathcal{L}_g$ improves the performance by over 5%, which verifies the effectiveness of leveraging the clean data as auxiliary supervision in feature learning. Also, it is observed that the image generation loss $\mathcal{L}_g$ contributes to the most performance gain as an individual module. It indicates that reconstructing the clean image, or preserving the text content, is the most important task when learning the feature representation.

Another interesting observation is that compared with image generation using the $\mathcal{L}_g$ loss only, adding the adversarial training in the image generation does not bring a significant improvement to the scene recognition performance. One possible reason may be revealed in the second example in Fig. 3, which has ground truth label 'coffee'. Although the image generated by the adversarial training looks more realistic than using $\mathcal{L}_g$ alone, as shown in Fig. 3, it interprets the last second character as 'l' instead of 'e', which leads to an incorrect prediction. This misunderstanding can be observed in both the generated image and the final prediction. Although using the image discriminator degrades the performance a little, it does provide us with a new possibility. With the help of the image discriminator, we can obtain a confidence score of the final prediction, which indicates the quality of generated images. The confidence score is close to 1 when a generated image looks realistic and to 0 otherwise. It is plotted in the last column of Fig. 3 for 25 local image regions from left to right. This confidence score has a correlation with character recognition accuracy and may be used in the lexicon-based word search. Since the image discriminator does not provide noticeable improvement in the recognition performance, in the following experiments, we disable the image discriminator unless otherwise specified.

On the other hand, adding the feature discriminator and adversarial training in the feature domain further boosts the recognition accuracy to 87%. It means that the adversarial training between the encoder and feature discriminator acts as a critical role in aligning the distribution between the features of the original

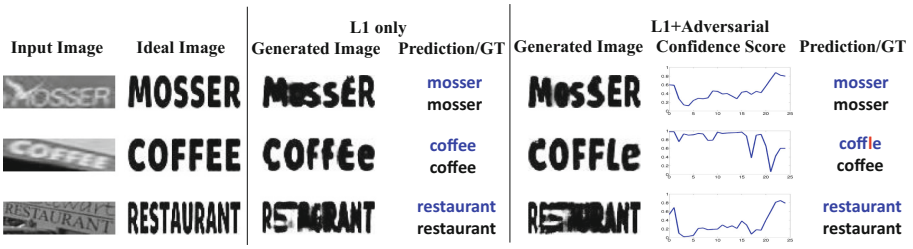


Fig. 3. Examples showing the generated images and their corresponding confidence scores. The first column shows the original input images and their paired 'clean' images. The middle column shows the generated images by using L_1 loss only and the corresponding prediction. The right column shows the generated images using L_1 loss with adversarial training, the corresponding confidence score and predictions. The confidence score corresponds to 25 local image regions which may contain one or part of a text glyph horizontally from left to right. The confidence score is close to 1 when it looks realistic and to 0 otherwise.

input image and the corresponding clean image. It makes the learned feature representation more exclusive or invariant to other nuisance factors.

4.3 Results and Comparisons on General Benchmarks

We evaluate our proposed method on the benchmarks that are designed for general scene text recognition, which mostly contain regular text although irregular text occasionally exists. The benchmark datasets are:

- **IIIT 5K-Words** [23]: (IIIT5K) contains 3000 cropped word images in its test set, which is collected from the Internet. Each image specifies a 50-word lexicon and a 1k-word lexicon.
- **Street View Text** [31]: (SVT) contains 647 test images, which are cropped from 249 Google Street View images. Many images in SVT suffer from severe noise and blur or have a very low resolution. Each image is associated with a 50-word lexicon.
- **ICDAR 2003** [21]: (IC03) contains 251 scene images labeled with text bounding boxes. For fair comparison [31], we discard the images contain non-alphanumeric characters or those having less than three characters. The resulting dataset contains 867 cropped images. Each cropped image is associated with a 50-word lexicon defined by Wang et al. [31] and a full lexicon which combines all lexicon words.
- **ICDAR 2013** [18]: (IC13) inherits most of its samples from IC03. After filtering samples as done in IC03, the dataset contains 857 samples. No lexicon is specified.

In Table 4, we report the performances of our synthetically-supervised feature learning model and compare them with 16 existing methods on the general

Table 4. Recognition rates (%) on standard scene text recognition benchmarks. '50' and '1k' refer to the lexicon sizes, 'Full' indicates the combined lexicon of all images in the benchmarks, and 'None' means unconstrained lexicon-free. Our method achieves the state-of-the-art performance across different benchmarks and significantly outperforms the state-of-the-art in the lexicon-free category.

Method	IIIT5K			SVT		IC03			IC13
	50	1K	None	50	None	50	Full	None	None
ABBEY [31]	24.3	-	-	35.0	-	56.0	55.0	-	-
SYNTH+PLEX [31]	-	-	-	57.0	-	76.0	62.0	-	-
Mishra et al. [10]	64.1	57.5	-	73.2	-	81.8	67.8	-	-
Wang et al. [32]	-	-	-	70.0	-	90.0	84.0	-	-
wDTW [6]	-	-	-	77.3	-	89.7	-	-	-
PhotoOCR [3]	-	-	-	90.4	78.0	-	-	-	87.6
Almazan et al. [1]	91.2	82.1	-	89.2	-	-	-	-	-
Strokelets [33]	80.2	69.3	-	75.9	-	88.5	80.3	-	-
Su and Lu [29]	-	-	-	83.0	-	92.0	82.0	-	-
Gordo [8]	93.3	86.6	-	91.8	-	-	-	-	-
Jaderberg et al. [17]	97.1	92.7	-	95.4	80.7	98.7	98.6	93.1	90.8
Jaderberg et al. [16]	95.5	89.6	-	93.2	71.7	97.8	97.0	89.6	81.8
CRNN [27]	97.6	94.4	78.2	96.4	80.8	98.7	97.6	89.4	86.7
RARE [28]	96.2	93.8	81.9	95.5	81.9	98.3	96.2	90.1	88.6
R^2AM [20]	96.8	94.4	78.4	96.3	80.7	97.9	97.0	88.7	90.0
FAN [4]	**99.3**	**97.5**	87.4	**97.1**	85.9	**99.2**	97.3	94.2	93.9
Ours	97.3	96.1	**89.4**	96.8	**87.1**	98.1	**97.5**	**94.7**	**94.0**

text recognition benchmarks. On unconstrained recognition tasks (recognizing without a lexicon), our method shows a significant improvement in all cases by using the clean image as supervision at both pixel level and at feature level in a generative way. More specifically, since CRNN [27] and our proposed method share the same encoder and text decoder network structure, thus it can serve as a strong baseline for fair comparison without adopting any auxiliary clean image for supervision. Our method outperforms CRNN by around 7% on average. This demonstrates the effectiveness and superiority of leveraging the auxiliary clean image. On constrained recognition tasks, we use a standard lexicon searching algorithm as in [27], and also achieve state-of-the-art or highly competitive results.

Compared with the method proposed in FAN [4], our method achieves competitive accuracies without using deep resent-based encoder or any attention mechanism as is done in FAN [4]. In addition, in the lexicon-free setting, our method significantly outperforms FAN on IIIT5K, SVT and performs comparably to the performance on IC03 and IC13. From our observations, we found that

Table 5. Recognition rates (%) on irregular text recognition benchmarks.

Method	SVT-Perspective	Curved text
Jaderberg et al. [17]	-	42.7
CRNN [27]	66.8	54.9
RARE [28]	71.8	59.2
Ours	**73.9**	**62.5**

IIIT5K and SVT contains more irregular text, especially curved text and has very low resolution images. Our method has an advantage in dealing with irregular text which have a large variance in their appearance. This may be because the learned text representation in our proposed method is largely invariant of the other nuisance factors, thus makes different text images are maximally distinguishable. In order to further verify the robustness and generalization capability of our proposed method, we provide more testing of our method on challenging irregular text recognition tasks in 4.4.

4.4 Results and Comparisons on Irregular Text Benchmarks

In this section, we evaluate our proposed algorithm on the irregular text scenarios to verify its effectiveness. We use the same model trained on the Synth90 dataset without fine-tuning. All models are evaluated without a lexicon. The two standard irregular text benchmark datasets are SVT-Perspective [25] and CUTE80 [26].

SVT-Perspective: (SVT-Perspective) contains 639 cropped images for testing, which is specially designed for evaluating the performance of perspective text recognition. Test samples are selected from side view angles in Google Street View. Thus most of them are heavily deformed by perspective distortion.

CUTE80: (CUTE80) contains 288 cropped word images for testing, which is collected from 80 high-resolution images taken in the natural scene. This dataset is specially designed for evaluating curved text recognition.

Table 5 summarizes the recognition performance on the SVT-Perspective and CUTE80 datasets. In comparison with other existing methods, which use the same training set, our method outperforms them by a large margin in all cases. Furthermore, recall the results in Table 4, compared with the baseline CRNN model, on SVT-Perspective our proposed method outperforms CRNN by an even larger margin than it does for SVT benchmark. The reason is that the SVT-Perspective dataset mainly consists of the perspective text, which is more challenging and inappropriate for direct recognition. Our synthetically-supervised feature learning can significantly alleviate this problem.

It worth noting that we achieve a significant improvement over RARE [28], which is a method designed specifically for irregular text. Our proposed model is simple and effective to address various kinds of irregular text in a unified way by learning from auxiliary 'clean' image. In addition, our methods does not need to detect the fiducial points and rectify the images before the image recognition procedure as is done in RARE. This further indicates that the learned text feature in our model is more robust to the variance of nuisance factors, i.e., the curved shape or the perspective angles. We also present some visual examples to compare the quality of the rectified images by RARE and generated image by our proposed method in Fig. 4. For given input examples listed in the first column, the second column represents the rectified images achieved by RARE, and the third column shows the generated images obtained by our image generator. We observe that our generated image is closer to a canonical view of the original input image, which eliminates off most of the appearance variance of the nuisance factors. In contrast to the RARE method, we do not use the generated image as a pre-processed step before the sequential text recognition. The generation of 'clean' image only aims to guide the feature learning.

In Fig. 5, we present some interesting examples to show some challenging and failure cases. Figure 5(a) shows some challenging examples our model makes a correct prediction, which shows that our model is robust to the undesired occlusion, background variation, geometric deformation in both the image generation and text decoding task. Figure 5(b) demonstrates some failure cases, which reveals that the prediction accuracy is always linked to the quality of the generated image closely. For instance, for the word 'phil', the second character in our generated image is similar to the character 'i', which is misclassified to 'i' in the prediction. For the input image 'i8', we predict its label to '18', which shows our model implicitly understands that numbers always appear together, which might comes from the bias in the training samples. In most cases, the predict result is consistent with the characters appears in the generated image. Most misclassified samples contain a short text or has a very low resolution.

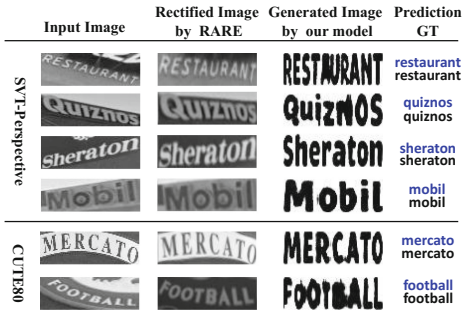
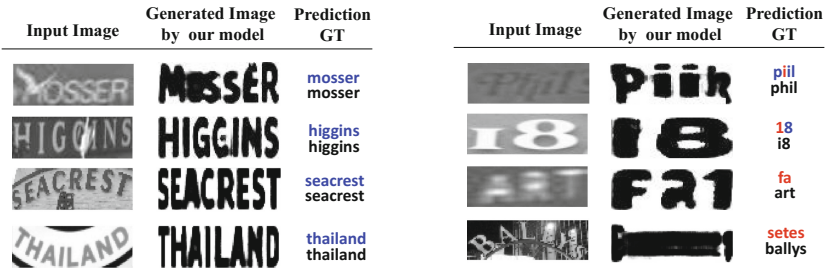


Fig. 4. Comparison of image rectification effects using our generative model versus the transformation model of RARE [28]. Our model correctly recognizes all these challenging examples.



(a) Successful Cases (b) Failure Cases

Fig. 5. Examples showing the images our model generated and the recognition results. In each sub-figure, the left column is the input images; the middle column is the generated image; the right column is the recognized text and the ground truth text. Blue and red characters are correctly and mistakenly recognized characters. (Color figure online)

5 Conclusions

We have presented a novel algorithm for scene text recognition. The core novelties of our method are the use of "clean" images that are readily available from the synthetic data generation process and a novel multi-task network with an encoder-generator-discriminator-decoder architecture that guides image feature learning by using clean images. We show that our method significantly outperforms the state-of-the-art methods on standard scene text recognition benchmarks. Furthermore, we show that without explicit handling, our method works on challenging cases where input images contain severe geometric distortion, such as text on a curved path. Future work might include studies on how different clean images may affect the performance of the recognition algorithm, how to use other parameters of the data generation process, such as font, as auxiliary data for feature learning, and how to train end-to-end systems that combine both text detection and recognition in this framework.

References

1. Almazán, J., Gordo, A., Fornés, A., Valveny, E.: Word spotting and recognition with embedded attributes. IEEE Trans. Pattern Anal. Mach. Intell. **36**(12), 2552–2566 (2014)
2. Berthelot, D., Schumm, T., Metz, L.: Began: boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717 (2017)
3. Bissacco, A., Cummins, M., Netzer, Y., Neven, H.: PhotoOCR: reading text in uncontrolled conditions. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 785–792. IEEE (2013)
4. Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., Zhou, S.: Focusing attention: towards accurate text recognition in natural images. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5086–5094. IEEE (2017)

5. Ghosh, S.K., Valveny, E., Bagdanov, A.D.: Visual attention models for scene text recognition. In: ICDAR (2017)
6. Goel, V., Mishra, A., Alahari, K., Jawahar, C.: Whole is greater than sum of parts: recognizing scene text words. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 398–402. IEEE (2013)
7. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
8. Gordo, A.: Supervised mid-level features for word image representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2956–2964 (2015)
9. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376. ACM (2006)
10. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. https://www.freetype.org
13. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arXiv preprint (2017)
14. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. In: NIPS Deep Learning Workshop (2014)
15. Jaderberg, M., Vedaldi, A., Zisserman, A.: Deep features for text spotting. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 512–528. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_34
16. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Deep structured output learning for unconstrained text recognition. In: International Conference on Learning Representations (2015)
17. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. Int. J. Comput. Vis. **116**(1), 1–20 (2016)
18. Karatzas, D., et al.: ICDAR 2013 robust reading competition. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 1484–1493. IEEE (2013)
19. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
20. Lee, C.Y., Osindero, S.: Recursive recurrent nets with attention modeling for ocr in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2231–2239 (2016)
21. Lucas, S.M., et al.: ICDAR 2003 robust reading competitions: entries, results, and future directions. Int. J. Doc. Anal. Recognit. (IJDAR) **7**(2–3), 105–122 (2005)
22. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
23. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: BMVC 2012–23rd British Machine Vision Conference. BMVA (2012)
24. Peng, X., Yu, X., Sohn, K., Metaxas, D.N., Chandraker, M.: Reconstruction-based disentanglement for pose-invariant face recognition. Intervals **20**, 12 (2017)

25. Phan, T.Q., Shivakumara, P., Tian, S., Tan, C.L.: Recognizing text with perspective distortion in natural scenes. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 569–576. IEEE (2013)
26. Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. Expert. Syst. Appl. **41**(18), 8027–8048 (2014)
27. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2017)
28. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4168–4176 (2016)
29. Su, B., Lu, S.: Accurate scene text recognition based on recurrent neural network. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9003, pp. 35–48. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16865-4_3
30. Tran, L., Yin, X., Liu, X.: Disentangled representation learning GAN for pose-invariant face recognition. In: CVPR (2017)
31. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1457–1464. IEEE (2011)
32. Wang, T., Wu, D.J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural networks. In: ICPR (2012)
33. Yao, C., Bai, X., Shi, B., Liu, W.: Strokelets: a learned multi-scale representation for scene text recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4042–4049 (2014)
34. Ye, Q., Doermann, D.: Text detection and recognition in imagery: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **37**(7), 1480–1500 (2015)
35. Yim, J., Jung, H., Yoo, B., Choi, C., Park, D., Kim, J.: Rotating your face using multi-task deep neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 676–684 (2015)
36. Yin, X.C., Zuo, Z.Y., Tian, S., Liu, C.L.: Text detection, tracking and recognition in video: a comprehensive survey. IEEE Trans. Image Process. **25**(6), 2752–2773 (2016)
37. Zhou, X., et al.: East: an efficient and accurate scene text detector. arXiv preprint arXiv:1704.03155 (2017)
38. Zhu, Y., Yao, C., Bai, X.: Scene text detection and recognition: recent advances and future trends. Front. Comput. Sci. **10**, 19–36 (2016)