SimAN: Exploring Self-Supervised Representation Learning of Scene Text via Similarity-Aware Normalization

Canjie Luo[1], Lianwen Jin[1,2,*], and Jingdong Chen[3]

[1]South China University of Technology,
[2]Peng Cheng Laboratory,
[3]Ant Group

Abstract

Recently self-supervised representation learning has drawn considerable attention from the scene text recognition community. Different from previous studies using contrastive learning, we tackle the issue from an alternative perspective, i.e., by formulating the representation learning scheme in a generative manner. Typically, the neighboring image patches among one text line tend to have similar styles, including the strokes, textures, colors, etc. Motivated by this common sense, we augment one image patch and use its neighboring patch as guidance to recover itself. Specifically, we propose a Similarity-Aware Normalization (SimAN) module to identify the different patterns and align the corresponding styles from the guiding patch. In this way, the network gains representation capability for distinguishing complex patterns such as messy strokes and cluttered backgrounds. Experiments show that the proposed SimAN significantly improves the representation quality and achieves promising performance. Moreover, we surprisingly find that our self-supervised generative network has impressive potential for data synthesis, text image editing, and font interpolation, which suggests that the proposed SimAN has a wide range of practical applications.

1 Introduction

To summarize, our contributions are as follows:

- We propose a generative (opposite of contrastive [34]) representation learning scheme by utilizing the unique properties of scene text, which might inspire rethinking the learning of better representations for sequential data like text images. To the best of our knowledge, this is the first attempt for scene text recognition.
- We propose a generative (opposite of contrastive [34]) representation learning scheme by utilizing the unique properties of scene text, which might inspire rethinking the learning of better representations for sequential data like text images. To the best of our knowledge, this is the first attempt for scene text recognition mented image patch and its neighboring patch to align corresponding styles. Only if the representations are sufficiently distinguishable, different patterns can be identified and be aligned with correct styles. Otherwise, the network might result in a wrong recovered image, e.g., in different colors.
- The proposed SimAN achieves promising representation performance. Moreover, the self-supervised network shows impressive capabilities to synthesize data, edit text images and interpolate fonts, suggesting the broad practical applications of the proposed approach.

3 Methodology

In this section, we first introduce the design of the pretext task and the construction of the training samples. Then, we detail the proposed SimAN module. Finally, we present the objectives of the task and the complete learning scheme. The overall framework is shown in Figure 2.

3.1 Training Sample Construction

Constructing appropriate training samples is critical to the success of the pretext task. We enable the scene text representation learning by recovering an augmented image patch using its neighboring patch as guidance. This design considers the unique properties of scene text, i.e., the styles (e.g., stroke width, textures, and colors) within one text line tend to be consistent.

The pretext task requires decoupled style and content inputs. As shown in Figure 2, given an unlabeled text image $I \in \mathbb{R}^{3\times H\times W}$ (the width W is required to be larger than two times of height H), we randomly crop two neighboring image patches $I_s, I_c \in \mathbb{R}^{3\times H\times H}$ as style and content input, respectively. This ensures sufficient differences in content between the two patches. Even if the neighboring patches might contain a same characters, their positions are different. Then, we augment (blurring, random noise, color changes, etc.) the content patch I_c as I_{aug} to make its style different from the style patch I_s. Finally, the pretext task takes I_{aug} as content input and I_s as the style guidance to recover an image I_{rec}. The source content patch I_c serves as supervision.

Discussion As our pretext task is recovering an augmented patch under the guidance of its neighboring patch, the visual cues should be consistent in both patches. Some

Table 3. Probe evaluation. We report the word accuracy (Acc., %) and word-level accuracy up to one edit distance (E.D. 1, %). The real training data provides more robust representations.

Probe	Traning Data		IIIT5K		IC03		IC13	
Type	Encoder	Probe	Acc.	E.D.1	Acc.	E.D.1	Acc.	E.D.1
CTC	Synth.	Synth.	60.8	75.6	64.9	78.9	64.0	81.0
	Real.	Synth.	**68.9**	**82.8**	**75.0**	**87.2**	**72.9**	**86.0**
Att.	Synth.	Synth.	66.5	78.8	71.7	83.6	68.7	81.6
	Real.	Synth.	**73.7**	**85.6**	**81.2**	**90.4**	**77.9**	**87.8**

Table 6. Arbitrary-length Text editing evaluation. We report FID score and word-level recognition accuracy (%). Although the supervised EditText can imitate more font category and background texture, our self-supervised approach achieves better readability.

Method	Supervision	FID↓	Acc.↑
EditText[57]	✓	**40.5**	14.9
Ours	×	67.9	**57.6**

spatial augmentation strategies, such as elastic transformation, might break the consistency and lead to failed training. For instance, it might bring changes to the stroke width. The excessively distorted strokes are also diverse from the source font style. Therefore, we avoid all of the spatial transformation augmentation methods that are widely used for self-supervised representation learning. This is also a significant difference with previous study SeqCLR [1].

3.2 Similarity-Aware Normalization

Previous studies [2, 3] revealed that the statistics of feature maps, including mean and variance, can represent styles. Based on this finding, we perform instance normalization (IN) [2, 4] on the feature maps to remove the style

$$\sigma_{c,i,j} = \frac{1}{3}\sqrt{\sum_{p,q\in\mathbb{N}_{i,j}} (x_{c,p,q} - \mu_{c,i,j})^2}, \tag{4}$$

3.3 Learning Scheme

As we formulate the pretext task as image reconstruction, the source patch Ic can serves as supervision. We minimize the distance between the recovered image Irec and target image Ic as

$$\mathcal{L}_2 = \| I_{rec} - Ic \|_2^2 . \tag{8}$$

Simultaneously, we adopt a widely used adversarial objective to minimize the distribution shift between the generated and real data:

$$\min_D \mathcal{L}adv = \mathbb{E}\left[(D(I_s) - 1)^2\right] + \mathbb{E}\left[(D(Irec))^2\right] \tag{9}$$

$$\min_{\text{Encoder, Decoder}} \mathcal{L}_{adv} = \mathbb{E}\left[(D(I_{rec}) - 1)^2\right] \tag{10}$$

where D denotes a discriminator. The complete learning scheme is shown in Algorithm 1. The encoder/decoder and discriminator are alternately optimized to achieve adversarial training

Algorithm 1 Representation Learning Scheme

Input: Encoder, Decoder, Discriminator D
Output: Encoder, Decoder
1: **for** iteration t = 0, 1, 2, ..., T **do**
2: Sample a mini-batch $\{\mathrm{I}_i\}_{i=1}^{B} fromunlabeleddata$
3: for each I_i**do**
4: Randomly crop $\mathrm{I}_s and I_c, augment I_c as I_{aug}$
5: Forward Encoder, SimAN and Decoder
6: Compute loss for $\{\mathrm{I}_{rec}, i\}_{i=1}^{B}$
7: Update D using $\min_D \mathcal{L}adv$
8: Update Encoder and Decoder using $\min_{\text{Encoder, Decoder}} \mathcal{L}_{adv}$
9: (The $\lambda isempirically set to 10$.)

4 Experiments

4.1 Dataset

4.2 Implementation Details

We provide more details, such as augmentations, architectures, probe objectives, and training settings, in the Supplementary Material.

Encoder/Decoder We adopt a popular recognizer backbone ResNet-29 [2] as our encoder. We symmetrically design a lightweight decoder.

Recognizer The complete architecture of the recognizer follows [1,2], including a rectification module, a ResNet-29 backbone, two stacked BiLSTMs and a CTC [15] /Attention [4] decoder, as shown in Figure 3.

Optimization In the self-supervised representation learning stage, we set the batch size to 256 and train the network for 400K iterations. It takes less than 3 days for convergence on two NVIDIA P100 GPUs (16GB memory per GPU). The optimizer is Adam [30] with the settings of $\beta_1 = 0.5 and \beta_2 = 0.999. The learning rate is set to 10^4 and linearly decreased to 10^5. The ima$

Acknowledgment
This research was supported in part by NSFC (Grant No. 61936003) and GD-NSF (No. 2017A030312006).

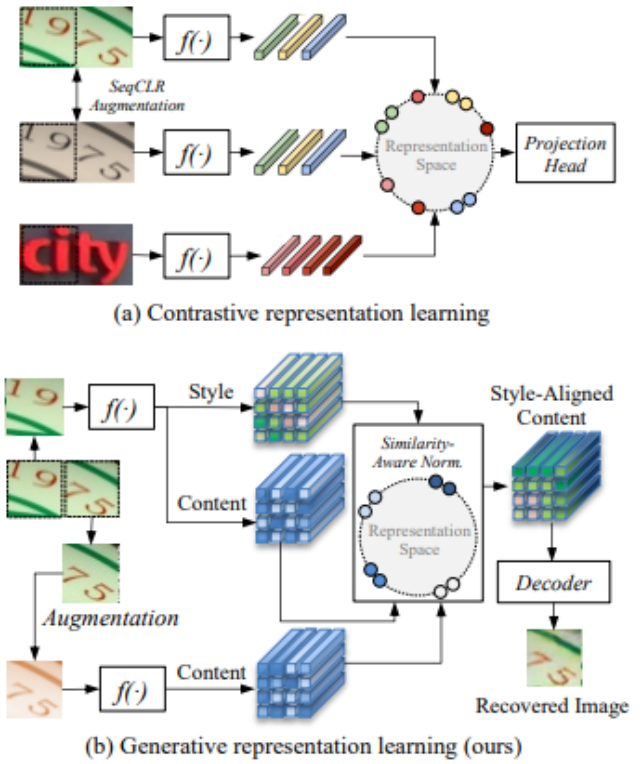

Figure 1: Scene text representation learning in (a) the contrastive and (b) the generative manner (ours). We estimate the similarity of the content representations between the augmented patch and its neighboring patch, and align the corresponding styles to reconstruct the augmented patch. Only high-quality representations are distinguishable so that a precise reconstruction can be achieved

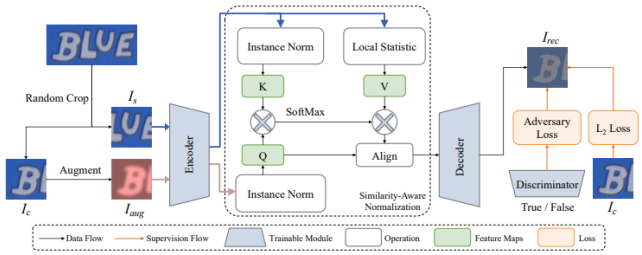

Figure 2: Overview of the proposed generative representation learning scheme. We decouple content and style as two different inputs and guide the network to recover the augmented image. The proposed SimAN module learns to align corresponding styles for different patterns according to the distinguishable representations.

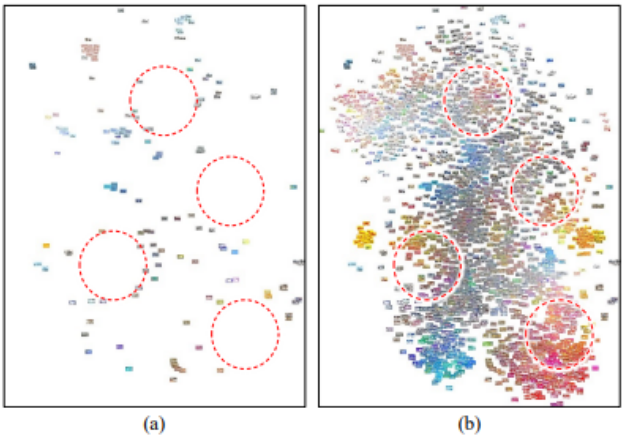

Figure 5: Distribution of scene text images containing the word "the" via t-SNE. We show two distributions of (a) 200 real labeled samples and (b) 200 real samples and our 2000 synthetic samples. The large empty space of original distribution might suggest the lack of diversity of labeled data. After adding our synthetic samples, the distribution is more even and dense. Best viewed in color

References

[1] Aviad Aberdam, Ron Litman, Shahar Tsiper, Oron Anschel, Ron Slossberg, Shai Mazor, R Manmatha, and Pietro Perona. Sequence-to-sequence contrastive learning for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15302–15312, 2021. 2

[2] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 2

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[4] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 2