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

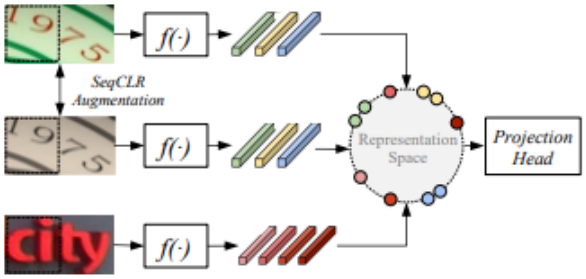
Canjie Luo[1], **Lianwen Jin** [1,2,*], and **Jingdong Chen**[3]

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

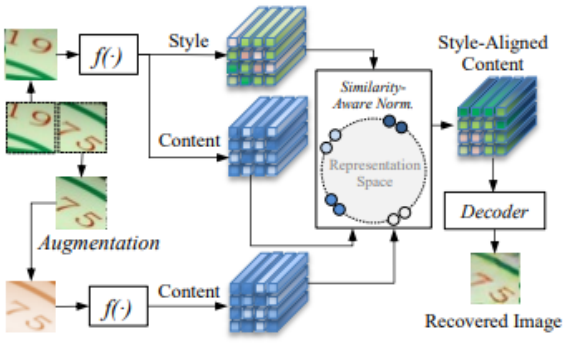
{canjie.luo, lianwen.jin}@gmail.com,jingdongchen.cjd@antgroup.com

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.



(a) Contrastive representation learning



(b) Generative representation learning (ours)

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

1 Introduction

The computer vision community has witnessed the great success of supervised learning over the last decade. However, the supervised learning methods heavily rely on labor-intensive and expensive annotations. Otherwise, they might suffer from generalization problems. Recently selfsupervised representation learning has become a promising alternative and is thus attracting growing interest [24,34]. It has been shown that the self-supervised representations can benefit subsequent supervised tasks [6–10, 18]. Despite the fast-paced improvements of representation learning on single object recognition/classification tasks

The learning scheme is shown in Figure 1(a). 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 SimAN module, which estimates the similarity of the representations between the augmented 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.

2 Related Work

2.1 Data Hunger of Scene Text Recognition

Scene text recognition is a crucial research topic in the computer vision community, because the text in images provides considerable semantic information for us. One important open issue in this field is data hunger. Typically, mainstream scene text recognizers [14, 45, 54] require a large number of annotated data. However, data collection and annotation cost a lot of resources. For instance, annotating a text string is tougher than selecting one option as the ground truth for single object classification datasets, whereas tens of millions of training data are required to gain robustness. Although synthetic data are available, previous studies [26,33,37,61] suggested that there is a gap between real and synthetic data. To mitigate this problem, Zhang et al. [61] and Kang et al. [26] proposed domain adaptation models to utilize unlabeled real data. Our study explores representation learning in a generative way, which is an alternative solution to make use of unlabeled real da

2.2 Visual Representation Learning

In the big data era, tremendous amounts of unlabeled data are available. Making the best use of unlabeled data becomes a crucial topic. Self-supervised representation learning has drawn massive attention owing to its excellent capability of pre-trained feature extraction [24, 34]. For instance, an encoder trained after a pretext task can extract transferrable features to benefit downstream tasks. We summarize popular methods into two main categories according to their objectives as follows. The contrastive learning scheme defines the pretext task as a classification task or a distance measuring task. For instance, the pretext task is to predict relative rotation [31] and position [56]. Recently the similarity measuring pretext task has become dominant, which aims to minimize the distance between the positive pairs while maximizing their distance to the negative ones using a discriminative head [5, 7, 8, 10, 18]. It is closely related to metric learning. Furthermore, the similarity measuring task using only positive pairs and discarding negative samples [9, 16] is also emerging topic. For the field of scene text, Baek et al. [3] introduced existing self-supervised techniques [18, 31] to use unlabeled data but resulted in approximately the same performance. Aberdam et al. [1] proposed a contrastive representation learning scheme, termed SeqCLR, to satisfy the sequence-to-sequence structure of scene text recognition. This is the first step towards scene text representation. The generative learning scheme has not been intensively studied in computer vision. One reason for this may be that the raw image signal is in a continuous and highdimensional space, unlike the natural language sentences in a discrete space (e.g., words or phrases) [18]. Therefore, it is difficult to define an instance. Although it is possible to model the image pixel by pixel [50], this theoretically requires much more high-performance clusters [6]. Another solution is the denoising auto-encoder [49,52], which learns features by reconstructing the (corrupted) input image. Our approach falls into the second category of visual representation learning, i.e., the generative learning scheme. We propose a novel representation learning scheme by studying the unique properties of scene text and using an image reconstruction pretext task

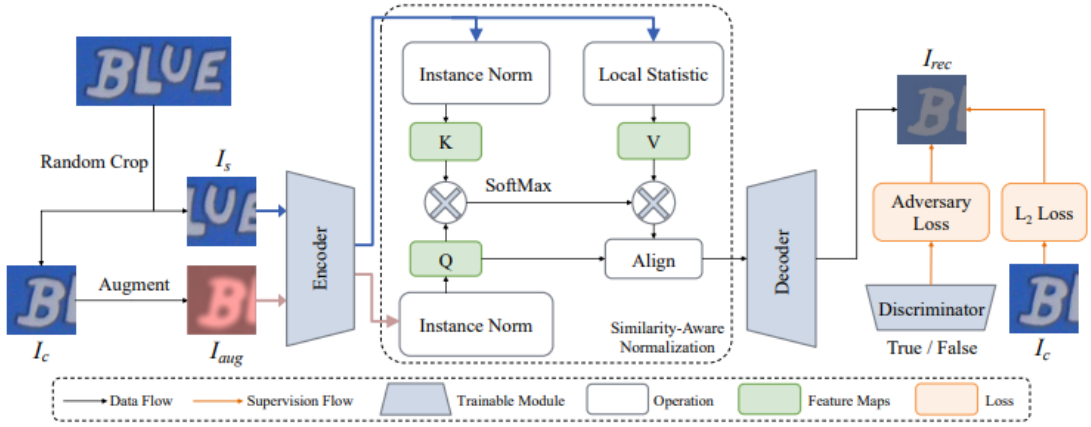


Figure 2: Overview of the proposed generative representation learning scheme. We decouple content and style as two different inputs and guide the network to recover the augmented image. The proposed SimAN module learns to align corresponding styles for different patterns according to the distinguishable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

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 R 3×H×W (the width W is required to be larger than two times of height H), we randomly crop two neighboring image patches Is, Ic R 3×H×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 Ic as Iaug to make its style different from the style patch Is. Finally, the pretext task takes Iaug as content input and Is as the style guidance to recover an image Irec. The source content patch Ic 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 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

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

3.3 Learning Scheme

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

$$\mathcal{L}_2 = \|I_{rec} - I_c\|_2^2 \qquad (2)$$

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

$$\min_D \mathcal{L}_{adv} = \mathbb{E}[(D(I_s) - 1)^2] + \mathbb{E}[(D(I_{rec}))^2], \qquad (3)$$

$$\min_{\text{Encoder, Decoder}} \mathcal{L}_{adv} = \mathbb{E}\big[(D(I_{rec}) - 1)^2\big], \quad (4)$$

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

Require: Encoder, Decoder, Discriminator D
Ensure: Encoder, Decoder
1: **for** iteration t = 0,1,2,...,T **do**
2: Sample a mini-batch $\{I_i\}_{i=1}^{B}$ from unlabeled data
3: **for** each I_i **do**
4: Randomly crop Is and Ic, augment Ic as Iaug
5: **end for**
6: Forward Encoder, SimAN and Decoder
7: Compute loss for $\{I_{rec,i}\}_{i=1}^{B}$
8: Update D using $\min_{D} \mathcal{L}_{adv}$
9: : Update Encoder and Decoder using
10: $\min_{\text{Encoder, Decoder}} \mathcal{L}_{adv} + \lambda \mathcal{L}_2$

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 104 and linearly decreased to 105 . The images are resized to a height of 32 pixels, maintaining the aspect ratio. The training setting of recognizers follows previous study SeqCLR [1].

4.3 Probe Evaluation

Moreover, we find that this experimental setting (pretraining the backbone and fine-tuning the probe using the very same synthetic dataset) might not meet the actual practice. In fact, we usually encounter one situation that we have vast amounts of unlabeled real-world data. It is worth making the best use of the real-world data. Therefore, we conduct an experiment under this new setting to further verify the effectiveness of our approach. We perform selfsupervised learning of the backbone using the Real-300K dataset. As shown in Table 1, the recognition performance is significantly boosted. As the real-world dataset provides more realistic and diverse images, it benefits the robustness of the backbone.

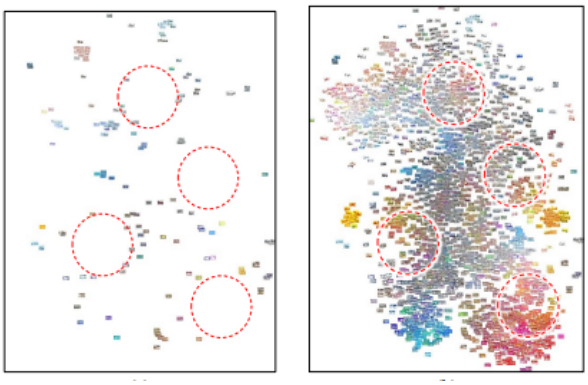
Table 1: Probe evaluation. We report the word accuracy (Acc., and word-level accuracy up to one edit distance (E.D. 1, real training data provides more robust representations.

Probe	Training 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.	**68.8**	**82.6**	**75.9**	**87.9**	**74.0**	**86.0**
Att.	Synth.	Synth.	66.8	78.6	71.9	83.9	68.6	81.7
	Real	Synth.	**73.8**	**85.6**	**81.9**	**90.9**	**77.0**	**87.8**

4.4 Semi-Supervision Evaluation

4.5 Generative Visual Tasks

4.5.1 Data Synthesis



(a) (b)

Figure 3: 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

First, we visualize the distributions of the limited real labeled samples and our plentiful synthetic samples. As shown in Figure 3, the limited labeled real-world data cannot cover diverse styles. However, our synthetic data fills the empty style space, indicating the significantly enriched styles. Then, we conduct

recognition experiments to show the quantitative results

4.5.2 Arbitrary-Length Text Editing

The goal of editing text in the wild is to change the word on the source image while retaining the realistic source look. As our approach can synthesize new words within source styles, we study the performance of our self-supervised approach and a popular supervised method EditText [57]. We generate 10K images using the corpus of SynthText [17] and the style of IC13 [28]. Then we evaluate the style distribution similarity using the FID score [20] and the readability using a mainstream recognizer3 [44]. As shown in Table 2, the EditText cannot handle target text of various lengths.

Table 2: 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	✓	**40.5**	14.9
Ours	×	67.9	**57.6**

Acknowledgement

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

References

[1] Aviad Aberdam **andothers**. "Sequence-to-sequence contrastive learning for text recognition". **in***Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*: 2021, **pages** 15302–15312.

[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*: 2017, **pages** 1501–1510.

[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*: 2019, **pages** 4401–4410.

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