Text Prior Guided Scene Text Image Super-Resolution

Jianqi Ma, Shi Guo, and Lei Zhang, *Fellow, IEEE*

***Abstract*— Scene text image super-resolution (STISR) aims to improve the resolution and visual quality of low-resolution (LR) scene text images, while simultaneously boost the performance of text recognition. However, most of the existing STISR methods regard text images as natural scene images, ignoring the categorical information of text. In this paper, we make an inspiring attempt to embed text recognition prior into STISR model. Specifically, we adopt the predicted character recognition probability sequence as the text prior, which can be obtained conveniently from a text recognition model. The text prior provides categorical guidance to recover high-resolution (HR) text images. On the other hand, the reconstructed HR image can refine the text prior in return. Finally, we present a multi-stage text prior guided super-resolution (TPGSR) framework for STISR. Our experiments on the benchmark TextZoom dataset show that TPGSR can not only effectively improve the visual quality of scene text images, but also significantly improve the text recognition accuracy over existing STISR methods. Our model trained on TextZoom also demonstrates certain generalization capability to the LR images in other datasets. The source code of our work is available at: https://github.com/mjq11302010044/TPGSR.**

***Index Terms*— Scene text image super-resolution, super-resolution, text prior.**

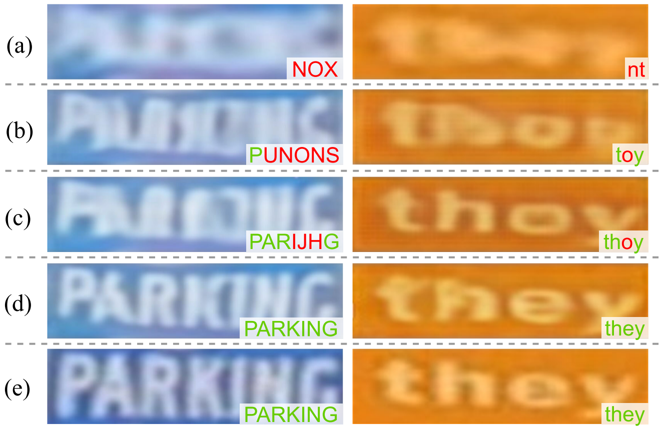


Fig. 1. Comparison of super-resolution results generated by (a) bicubic, (b) TSRN [5], (c) TSRN with fixed text prior, (d) TSRN with refined text prior and (e) HR. The bottom-right corner of the image shows the text recognition results.

I. Introduction

SCENE text image recognition aims to recognize the text characters from the input image, which is an important computer vision task that involves text information processing. It has been widely used in text retrieval [1], sign recognition [2], license plate recognition [3] and other scene-text-based visual question answering [4]. However, due to various issues such as low sensor resolution, blurring, poor illumination, etc., the quality of captured scene text images may not be good enough, which brings many difficulties to scene text recognition in practice. In particular, scene text recognition from low-resolution (LR) images remains a challenging problem.

In recent years, single image super-resolution (SISR) techniques have achieved significant progress owing to the rapid development of deep neural networks. Dong et al. [6] introduced hourglass structure into SISR task to progressively improve the SR performance. Ledig et al. [7] proposed to generate realistic SR outputs with GAN-based supervision. Lim et al. [8] adapted the residual network structure to SISR model to strengthen the image feature for superior SR performance. Zhang et al. [9] further upgraded the performance of SISR model with the channel attention. Inspired by the success of SISR, researchers have started to investigate scene text image super-resolution (STISR) to improve the quality of LR text images and hence improve the text recognition accuracy. Tran et al. [10] adapted LapSRN [11] to STISR and significantly improved the text content details. To obtain more realistic STISR results, Bílková et al. [12] and Wang et al. [13] employed GAN based networks with CTC loss [14] and text perceptual losses. In these methods, the LR images are synthesized (*e.g.*, bicubic down-sampling) from high resolution (HR) images for SR model learning, whereas the image degradation process in real-world LR images can be much more complex. Recently, Wang et al. [5] collected a real-world STISR dataset, namely TextZoom, where LR-HR image pairs captured by zooming lens are provided. Wang et al. also proposed a TSRN model for STISR, achieving state-of-the-art performance [5].

However, the existing STISR methods, *e.g.*, TSRN [5], mostly treat scene text images as natural scene images to perform super-resolution, ignoring the important semantic categorical information brought by the text content in the

Manuscript received 14 September 2021; revised 12 March 2022 and 1 September 2022; accepted 31 December 2022. Date of publication 10 February 2023; date of current version 22 February 2023. This work was supported by the Hong Kong Research Grants Council (RGC) Research Impact Fund (RIF) Grant R5001-18. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wangmeng Zuo. *(Corresponding author: Lei Zhang.)*

The authors are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csjma@comp.polyu.edu.hk; csshiguo@comp.polyu.edu.hk; cslzhang@comp.polyu.edu.hk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2023.3237002, provided by the authors.

Digital Object Identifier 10.1109/TIP.2023.3237002

image. As shown in Fig. 1(b), the result by TSRN [5] is much better than the simple bicubic model (Fig. 1(a)), but it is still hard to tell the characters therein. Based on the observation that semantic information can help to recover the shape and texture of objects [15], in this paper we propose a new STISR method, namely text prior guided super-resolution (TPGSR), by embedding the text recognition prior information into SR generation. Unlike the face segmentation prior used in [16] and the semantic segmentation used in [15], the text character segmentation is hard to obtain and there are few datasets containing annotations of fine character segmentation masks. We instead employ a text recognition model (*e.g.*, CRNN [17]) to extract the probability sequence of the text as the categorical prior of the given LR scene text image. Compared with other semantic prior information such as segmentation maps [15], the text recognition prior is coarser-grained, which is hard to be directly activated in SR models. Therefore, we adopt a TP transformer module to transform the coarse-grained prior guidance into finer-grained image feature priors, and embed it into the super-resolution network to guide the reconstruction of HR images. As can be seen in Fig. 1(c), the text prior information can indeed improve much the STISR results, making the text characters much more readable. On the other hand, the reconstructed HR text image can be used to refine the text prior, and consequently a multi-stage TPGSR framework can be built for effective STISR. Fig. 1(d) shows the super-resolved text image by using the refined text prior, where the text can be clearly and correctly recognized. The major contributions of our work are as follows:

- We, for the first time to our best knowledge, successfully introduce the text recognition categorical probability sequence as the semantic prior for STISR task, and validate its effectiveness to improve the visual quality and recognition accuracy of scene text images.
- We propose to refine the text recognition categorical prior without using extra supervision except for the real HR image, *i.e.*, refining recurrently by the estimated HR image and by fine-tuning the text prior generator with our proposed TP Loss. With such refinement, the text prior and the super-resolved text image can be jointly enhanced under our TPGSR framework.
- By improving the image quality of LR text images, the proposed TPGSR improves the text recognition performance on TextZoom for different text recognition models by a large margin and demonstrates good generalization performance to other recognition datasets.

II. Related Works

A. Single Image Super Resolution (SISR)

Aiming at estimating a high-resolution (HR) image from its low-resolution (LR) counterpart, SISR is a highly ill-posed problem [18]. In the past, handcrafted image priors are commonly used to regularize the SISR model to improve the image quality. In recent years, deep neural networks (DNNs) have dominated the research of SISR. The pioneer work SRCNN [19] learns a three-layer convolutional neural network (CNN) for the SISR task. Later on, many deeper CNN models have been proposed to improve the SISR performance, *e.g.*, deep residual block [8], Laplacian pyramid structure [11], densely connected network [20] and channel attention mechanism [9]. To better handle real-world SISR problems, Xu et al. [21] tried to reconstruct high-resolution RGB images from the sensor raw data, and Shocher et al. [22] adopted internal learning techniques to adapt the test image in unknown conditions. The PSNR and SSIM [23] losses are widely used in those works to train the SISR model. In order to produce perceptually-realistic SISR results, SRGAN [7] employs a generative adversarial network (GAN) to synthesize image details. SFT-GAN [15] utilizes the GAN loss and FSR-Net [16] employs semantic segmentation to generate visually pleasing HR images.

B. Scene Text Image Super Resolution (STISR)

Different from the general purpose SISR that works on natural scene images, STISR focuses on scene text images, aiming to improve the readability of texts by improving their visual quality. Intuitively, those methods for SISR can be directly adopted for STISR. In [24], Dong et al. extended SRCNN [19] to text images, and obtained the best performance in ICDAR 2015 competition [25]. PlugNet [26] employs a light-weight pluggable super-resolution unit to deal with LR images in feature domain. TextSR [13] utilizes the text recognition loss and text perceptual loss to generate the desired HR images for text recognition. To improve the performance of STISR on real-world scene text images, Wang et al. [5] built a real-world STISR image dataset, namely TextZoom, where the LR and HR text image pairs were cropped from real-world SISR datasets [27], [28]. They also proposed a TSRN [5] method to use the central alignment module and sequential residual block to exploit the semantic information in internal features. SCGAN [29] employs a multi-class GAN loss as supervision to equip the model with ability to generate more distinguishable face and text images.

By progressively adopting the high-frequency information derived from the text image, Quan et al. [30] proposed a multi-stage model for recovering blurry text images in high-frequency and RGB domain collaboratively. By adopting a recognition model as the perceptual unit, TBSRN [31] aims to supervise the SR text image generation at the content level in the learning process. PCAN [32] introduces the channel attention to the SR blocks and further upgrades the SR recovery performance. All the above methods estimate the HR text image by inputting LR image or low-level domain information, ignoring that the high-level text recognition categorical prior can also be an efficient guidance embedding.

C. Scene Text Recognition

In the early stage of deep learning based scene text recognition, researchers intended to solve the problem in a bottom-up manner [33], [34], *i.e.*, extracting the text from characters into words. Some other approaches recognize the text in a top-down fashion [35], *i.e.*, regarding the text image as a whole and performing a word-level classification. Taking text

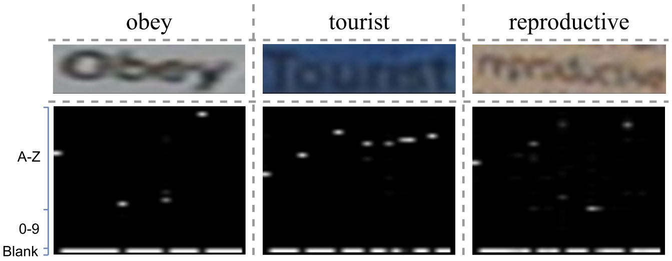


Fig. 2. Visualization of the text prior (TP) of some text images. The top, middle and bottom rows present the text labels, input text images and their TPs, respectively. From bottom to top, the categories of TP are ordered as the blank, 0-9 and A-Z, respectively. Zoom in for better visualization.

recognition as an image-to-sequence problem, CRNN [17] employs the CNN to extract image features and uses the recurrent neural networks to model the semantic information of image features. Trained with the CTC [14] loss, the predicted sequence can be more accurately aligned with the target sequence [36]. Recently, attention-based methods thrive due to their improvement in text recognition benchmarks and the robustness to various shapes of text images [37], [38]. In our method, we adopt CRNN as the text prior generator to generate categorical text priors for STISR model training. It is shown that such text priors can significantly improve the perceptual quality of super-resolved text images and consequently boost the text recognition performance.

III. Methodology

In this section, we will first explain what the text prior (TP) is, and then introduce the text prior guided super-resolution (TPGSR) network in detail, followed by the design of loss function.

A. Text Prior

In this paper, the TP is defined as the deep categorical representation of a scene text image generated by some text recognition models. The TP is then used as guidance information to encourage our TPGSR network to produce high-quality scene text images, which are favorable to both visual perception and scene text recognition.

Specifically, we choose the classic CTC-based text recognition model CRNN [17] to be the TP Generator. CRNN uses several convolution layers to extract the text features and five max pooling layers to down-sample the features into a feature sequence. The TP is then defined as the categorical probability prediction by CRNN, which is a sequence of $|A|$-dimensional probability vectors ($|A|$ denotes the number of characters learned by CRNN). Fig. 2 visualizes the TP of some scene text images, where the horizontal axis represents the sequence in left-to-right order and the vertical axis represents the categories in an alphabet order from bottom to top (*i.e.*, the blank, '0' to '9' and 'A' to 'Z'). In the visualization, the lighter the spot is, the higher the probability of this category will be. The CRNN recognition predictions are presented by order corresponding to the location of characters in the input image. The category probability and location alignment information make the TP an effective guidance to the SR generation. By using the TP as guidance, our TPGSR model can recover visually more pleasing HR images with higher text recognition accuracy, as we illustrated in Fig. 1.

B. The Architecture of TPGSR

By introducing TP into the STISR process, the main architecture of our TPGSR is illustrated in Fig. 3. Our TPGSR network has two branches: a TP generation branch and a super-resolution (SR) branch. First, the TP branch intakes the LR image to generate the TP feature. Then, the SR branch intakes the LR image and the TP feature to estimate the HR image. In the following, we introduce the two branches in detail.

1) TP Generation Branch: This branch uses the input LR image to generate the TP feature and passes it to the SR branch as guidance for more accurate STISR results. The key component of this branch is the TP Module, which consists of a learnable TP Generator and a TP transformer module. As mentioned in Section III-A, the TP generated by TP Generator is a probability sequence, whose size may not match the image feature map in the SR branch. To solve this problem, we employ a TP transformer module to transform the TP sequence into a feature map.

Specifically, the input LR image is first resized to match the input of TP Generator by bicubic interpolator, and then passed to the TP Generator to generate a TP matrix whose width is L. Each position of the TP is a vector of size $|A|$, which is the number of categories of alphabet A adopted in the recognizer. To align the size of TP feature with the size of image feature, we pass the TP feature to the TP transformer module. The TP transformer module consists of 4 Deconv blocks, each of which consists of a deconvolution layer, a BN layer and a ReLU layer. For an input TP matrix with width L and height $|A|$, the output of TP transformer module will be a feature map with recovered spatial dimension and channel C_T (usually 32) after three deconvolution layers with stride $(2, 2)$ and one deconvolution layer with stride $(2, 1)$. The kernel size of all deconvolution layers is 3×3.

2) SR Branch: The SR branch aims to reproduce an HR text image from the input LR image and TP guidance feature. It is mainly composed of an SR Module. Many of the SR blocks in existing SISR methods (*e.g.*, residual blocks [7], [20], enhanced residual blocks [8]) and STISR methods (*e.g.*, sequential-recurrent blocks [5]) can be adopted as our SR Module in couple of our TP guidance features. Considering that these SR blocks, such as the residual block in SRResNet [7] and the sequential-recurrent blocks in TSRN [5], only take the image features as input, we need to modify them in order to embed the TP features. We call our modified SR blocks as TP-Guided SR Blocks.

The difference between previous SR Blocks and our TP-Guided SR Block is illustrated in Fig. 4. To embed the TP features into the SR Block, we concatenate them to the internal image features along the channel dimension. Before the concatenation, we align the spatial size of TP features to that of the image features by bicubic interpolation. Suppose that the channel number of image features is C, then the

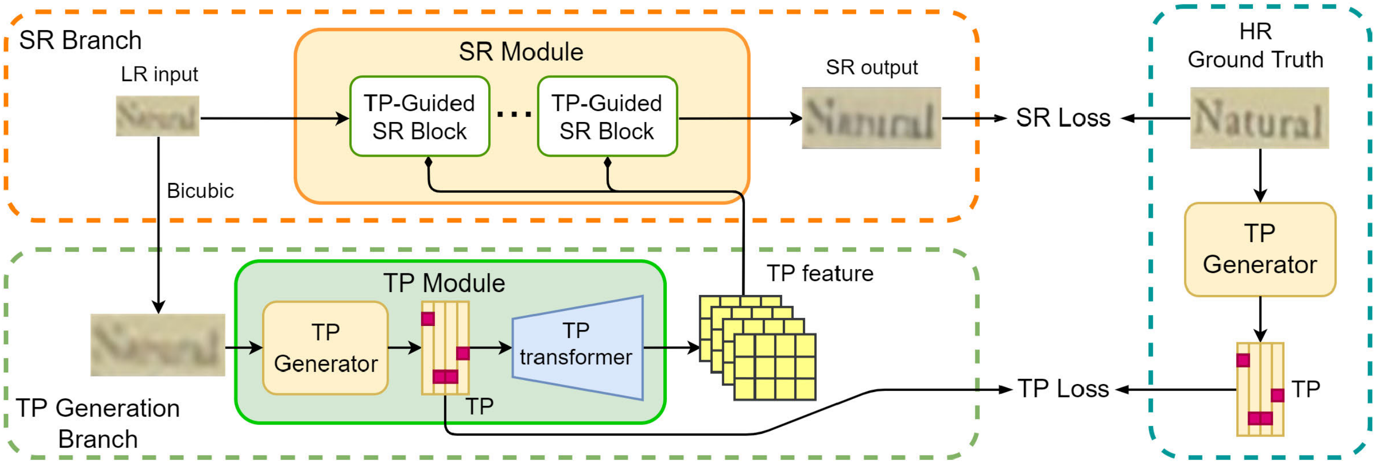


Fig. 3. Our proposed TPGSR framework, which consists of a Text Prior Generation Branch and a Super-resolution (SR) Branch. Accordingly, TP loss and SR loss are employed to train the whole network.

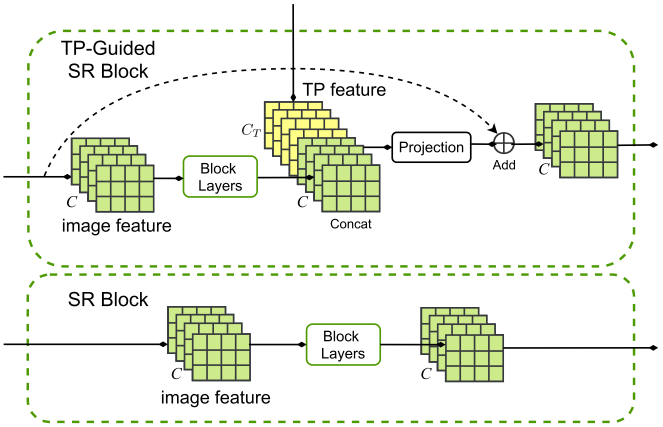


Fig. 4. Comparison of our TP-Guided SR Block and a common SR Block. In each block, the channel numbers of image features and TP features are C and C_T, respectively.

concatenated features of $C + C_T$ channels will go through a projection layer to reduce the channel number back to C. We simply use a 1×1 kernel convolution to perform this projection. The output of projection layer is fused with the input image feature by addition. With several such TP-Guided SR Blocks, the SR branch will output the estimated HR image, as in those previous super-resolution models.

C. Multi-Stage Refinement

With the TPGSR framework described in Section III-B, we can super-resolve an LR image to a better quality HR image with the help of TP features extracted from the LR input. One intuitive question is, if we extract the TP features from the super-resolved HR image, can we use those better quality TP features to further improve the super-resolution results? Actually, multi-stage refinement has been widely adopted in many computer vision tasks such as object detection [39] and instance segmentation [40] to improve the prediction quality progressively. Therefore, we extend our one-stage TPGSR model to a multi-stage learning framework by passing the estimated HR text image in one stage to the TP Generator in next stage. The multi-stage TPGSR framework is illustrated in Fig. 5. In the 1st stage, the TP Module accepts the bicubically interpolated LR image as input, while in the following stages, the TP Module accepts the HR image output from the SR Module in previous stage as input for refinement. As we will show in the ablation study in Section IV-B, both the quality of estimated HR text image and the text recognition accuracy can be progressively improved by this multi-stage refinement.

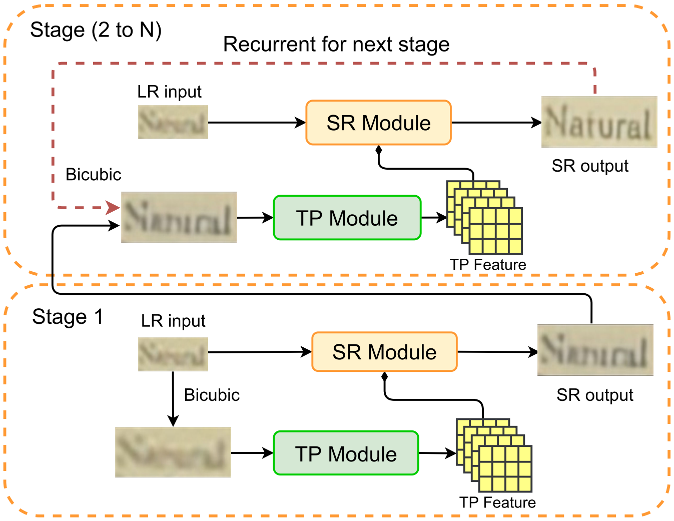


Fig. 5. Illustration of multi-stage TPGSR. The super-resolution output of one stage will be the text image input of next stage.

D. Training Loss

As shown in Fig. 3, there are two types of loss functions in our TPGSR framework, one for the SR branch and another for the TP generation branch. For the SR branch, the loss is similar to that in many previous SISR methods (*e.g.*, Lai et al. [11], Lim et al. [8] and Zhang et al. [20]). Denote by $\hat{I}_H$ the estimated HR image from the LR input and by I_H the ground-truth HR image, the loss for the SR branch, denoted by L_S, can be commonly defined as the L_1 norm distance between

$\hat{I}_H$ and I_H, *i.e.*,

$$L_S = |\hat{I}_H - I_H|. \tag{1}$$

Different from the many SISR works [8], [11], [20] as well as the STISR works [5], [13], in TPGSR we have loss functions specifically designed for the TP generation branch, which is crucial to improve the text image quality and text recognition. The TP sequence generated by the TP Generator has significant impact on the final SR results. More informative the TP is (*i.e.*, high probability at correct category), more positive impact it will bring on the estimated HR image. Let's use an example to illustrate the role of TP. In the first column of Fig. 6 (Fig. 6(a)), we show the SR result of an input LR image without using TP (*i.e.*, the result of traditional SR). One can see that the obtained HR image is not clear and there are some semantic errors. In the second column of Fig. 6 (Fig. 6(b)), we show another extreme, *i.e.*, we use the ground-truth HR image to extract the TP, and input it to the SR module for image recovery. One can see that a very clear and semantically correct SR image can be obtained. This validates that an accurate TP can help the SR of text images a lot. In Fig. 6(c), we extract the TP from the original LR image, and fix this TP to the SR module for text image recovery. One can see that though the extracted TP is not very close to the TP extracted from the HR image, it does provide useful information to aid SR, and the output SR image is much better than that without TP (Fig. 6(a)). Some of the characters can be correctly recognized. In inference, the HR image is not available and we can only estimate the TP from the LR image or the SR image (in multi-stage model). Intuitively, if we can train the model so that the TP generated from the LR image can approach to the TP extracted from the HR image, better SR results can be anticipated.

Based on the above discussion, we denote by t_L and t_H the TPs extracted from LR image I_L and HR ground-truth image I_H, respectively, and use the L_1 norm distance $|t_H - t_L|$ and the KL divergence $D_{KL}(t_L||t_H)$ to measure the similarity between t_L and t_H. With the text priors $t_L, t_H \in \mathcal{R}^{L\times|A|}$ of the pair of LR and HR images, the $D_{KL}(t_L||t_H)$ can be calculated as follows:

$$D_{KL}(t_L||t_H) = \sum_{i=1}^{L}\sum_{j=1}^{|A|} t_H^{ij} \ln \frac{t_H^{ij} + \epsilon}{t_L^{ij} + \epsilon}, \tag{2}$$

where t_L^{ij} and t_H^{ij} denote the element in the ith position and the jth dimension in t_L and t_H. ϵ is a small positive number to avoid numeric error in division and logarithm. Together with L_S, the overall loss function for a single-stage TPGSR can be written as follows:

$$L = L_S + \alpha\ |t_H - t_L| + \beta\ D_{KL}(t_L||t_H), \tag{3}$$

where α and β are the balancing parameters. With this loss, the TP Generator can be tuned to generate better TP that is more similar to that of the HR image. As shown in the rightmost column of Fig. 6 (Fig. 6(d)), the TP generated by the fine-tuned TP module is very close to that of the HR image (Fig. 6(b)), and the SR result is much better than Fig. 6(c), where the TP is extracted from a fixed TP Generator.

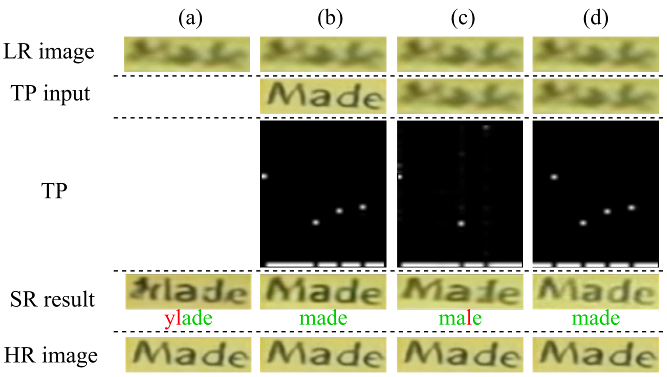


Fig. 6. Visualization of different TPs and the SR results. From top row to bottom row are the input LR image, TP Module input, generated TP, SR result and ground-truth HR image. From (a) to (d), we show the SR results without TP (*i.e.*, TSRN [5]), with TP generated from the HR mage, with TP generated from the LR image using fixed TP Generator, with TP generated from LR image using fine-tuned TP Generator.

For the multi-stage TPGSR learning, the loss for each stage, denoted by L_i, can be similarly defined as in Eq. 3. Suppose there are N stages in total, the overall loss is defined as follows:

$$L_{mt} = \sum_{i=1}^{N} \lambda_i\ L_i, \tag{4}$$

where λ_i balances the loss of each stage and $\sum_{i=1}^{N} \lambda_i = 1$.

E. Implementation Details

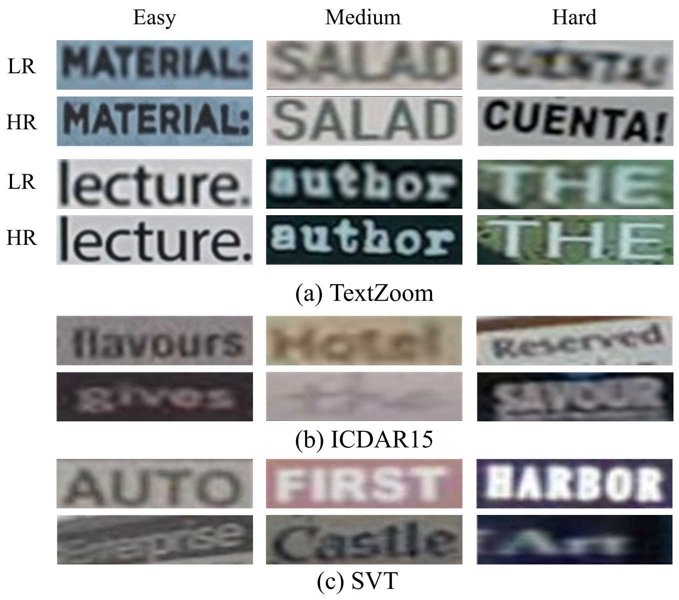
We implement our TPGSR method in PyTorch 1.2. Adam is selected as our optimizer with momentum 0.9. The batch size is set to 48 and the model is trained for 500 epochs with one NVIDIA RTX 2080Ti GPU. CRNN [17] pre-trained on SynthText [41] and MJSynth [42] is selected as the TP Generator. In Eq. 3, the weights α and β are both simply set to 1, while the ϵ in Eq. 2 is set to 10^{-6}. The alphabet set A includes mainly alphanumeric (0 to 9 and 'a' to 'z') case-insensitive characters. Together with a blank label, $|A|$ (*i.e.*, the size of A) has 37 categories in total. For dealing with the out-of-category cases, we assign all out-of-category characters with blank label, and the reconstruction of these characters mainly depends on the SR Module.

For multi-stage TPGSR training, we adopt a well-trained single-stage model to initialize all stages and cut the gradient across stages to speed up the training process to converge. The TP Generator are non-shared while the SR Module are shared cross stages. As in previous multi-stage learning methods [43], higher weight is assigned to the loss on the last stage, and the other stages are assigned with smaller weights on loss. In particular, we use a 3-stage TPGSR. The parameters λ_i in Eq. 4 are set as $\lambda_1 = \frac{1}{4}$, $\lambda_2 = \frac{1}{4}$ and $\lambda_3 = \frac{1}{2}$.

IV. Experiments

A. Datasets and Experiment Settings

1) Datasets: The TextZoom [5], ICDAR2015 [44] and SVT [45] datasets are used to validate the effectiveness of our



(a) TextZoom

(b) ICDAR15

(c) SVT

Fig. 7. Sample images from (a) TextZoom, (b) ICDAR15 and (c) SVT.

TABLE I

Ablation on Tuning the TP Generator. L_1 and D_{KL} Mean the L_1 Norm Distance and the KL Divergence. '✓' and '×' Mean Whether the TP Generator Is Fine-Tuned With the Corresponding Loss or Not. ACC Means the Average Recognition Accuracy

Approach	L_1	D_{KL}	ACC
TSRN [5]	-	-	41.4%
TPGSR-TSRN	×	×	44.5%
TPGSR-TSRN	✓	×	47.8%
TPGSR-TSRN	×	✓	48.9%
TPGSR-TSRN	✓	✓	**49.8%**
HR	-	-	72.3%

TABLE II

Ablation on Different α and β. ACC Means the Average Recognition Accuracy

α/β	ACC	β/α	ACC
1	**49.8%**	1	**49.8%**
2	48.9%	2	49.3%
5	48.6%	5	48.7%
10	48.5%	10	48.5%
20	48.2%	20	48.0%

proposed TPGSR method. Sample images of the three datasets are shown in Fig. 7.

TextZoom consists of 21,740 LR-HR text image pairs collected by lens zooming of the camera in real-world scenarios. The training set has 17,367 pairs, while the test set is divided into three subsets based on the camera focal length, namely easy (1,619 samples), medium (1,411 samples) and hard (1,343 samples). Some image pairs are shown in Fig. 7(a). The dataset also provides the text label for each pair.

ICDAR2015 is a well-known scene text recognition dataset, which contains 2,077 cropped text images from street view photos for testing. Since the images are captured incidentally on the street, the text images suffer from low resolution and blurring, making the text recognition very challenging. Some sample images are shown in (Fig. 7(b)).

SVT is also a scene text recognition dataset, which contains 647 testing text images. Each image has a 50-word lexicon with it. The images are also captured in the street and have low-quality, as shown in Fig. 7(c).

2) Experiment Settings: Since there are real-world LR-HR image pairs in the TextZoom dataset, we first use it to train and evaluate the proposed TPGSR model. We then apply the trained model to ICDAR2015/SVT to test its generalization performance to other datasets. Considering the fact that most of the images in ICDAR2015 and SVT have good resolution and quality, while the TextZoom training data focus on LR images, we perform the generalization test only to the low quality images in ICDAR2015/SVT whose height is less than 16 or the recognition score is less than 0.9.

B. Ablation Studies

To better understand the proposed TPGSR model, in this section we conduct a series of ablation experiments on the selection of parameters in loss function, the selection of number of stages and whether the TP Generator should be fine-tuned in training. We also perform experiments to validate the effectiveness of SR Module in our TPGSR framework. We adopt TSRN [5] as the SR Module in the experiments, and name our model as TPGSR-TSRN. All ablation experiments are performed on TextZoom and the recognition accuracies are evaluated with CRNN [17].

1) Impact of Tuning the TP Generator: To prove the significance of TP Generator tuning, we conduct experiments by fixing and tuning the TP Generator in a one-stage TPGSR model. The text recognition accuracies are shown in Table I. By fixing the TP Generator, we can enhance the SR image recognition by 3.1% compared to the TSRN baseline [5]. By tuning the TP Generator with the full set of loss (in Eq. 3) during the training process, the recognition accuracy can be further improved from 44.5% to 49.8%, achieving a performance gain of 5.3%. This clearly demonstrates the benefits of tuning the TP Generator to the SR text recognition task. However, if we disable L_1 or D_{KL} in Eq. 3 when tuning the TP Generator, the performance drops by 0.9% (49.8% *v.s.* 48.9%) or 2.0% (49.8% *v.s.* 47.8%) compared to the full loss training. The results reveal that the two loss terms are complementary with each other. They measure the similarity from different aspects and hence enrich the supervision information when used together.

2) Selection of Balancing Parameters in Loss: Referring to Eq. 3, there are two parameters, α and β, to balance the three parts in our loss function. We conduct experiments by using a single-stage TPGSR-TSRN to investigate how the different proportions of α and β affect the final SR text recognition accuracy. The experimental results are shown in Table II. We fix α to 1 and evaluate the models trained with different α/β ratios (ranging from 1 to 20). The model achieves the best result when β is set to 1 (left part of Table II). Then we fix β to 1 and search for the best β/α. The results in the right part of Table II reveal that the model gets the best text recognition accuracy of 49.8% when the ratio is set to 1. Therefore, we set both α and β to 1 in our experiments.

TABLE III

Ablation on Different Stage Settings. ‘E’, ’M’ and ‘H’ Mean the Accuracies of ‘Easy’, ’Medium’ and ‘Hard’ Split in TextZoom. ACC Means the Average Recognition Accuracy

N	E	M	H	ACC
1	61.0%	49.9%	36.7%	49.8%
2	62.2%	51.3%	37.4%	50.9%
3	63.1%	52.0%	38.6%	51.8%
4	63.7%	53.3%	**39.4%**	52.6%
5	**64.3%**	**54.2%**	39.2%	**53.1%**

TABLE IV

Ablation on Different Sharing Strategies. ‘✓’ Means That the Weights Are Shared in All Stages, While ‘×’ Means That the Weights Are Independent in Different Stages. ‘TP’ and ‘SR’ Mean the TP Module and SR Module in TPGSR. ACC Means the Average Recognition Accuracy

stage(N)	TP	SR	ACC
1	✓	×	49.8%
2	×	✓	50.9%
3	×	✓	**51.8%**
2	×	×	50.2%
3	×	×	51.5%
3	✓	✓	49.2%
3	✓	×	49.2%

TABLE V

Ablation on the Impact of SR. I_L and $\hat{I}_H$ Refer to Using the LR Image and Final SR Image as Input to the Recognizer. ACC Means the Average Recognition Accuracy, While ACC$_T$ Means the Average Recognition Accuracy With Tuned TP Generator

	ACC		ACC$_T$	
N	I_L	$\hat{I}_H$	I_L	$\hat{I}_H$
1	26.8%	49.8%	**45.3%**	50.5%
2	26.8%	50.9%	43.6%	51.6%
3	26.8%	**51.8%**	43.1%	**52.9%**

TABLE VI

SR Recognition Results by CRNN [17] and Image Quality Evaluation (PSNR/SSIM) on Different Stage Settings on TextZoom. N Refers to the Stage Number. ‘L’ Means That the Inputs of SR Branch in All Stages Are the Original LR Image and the SR Branch Is Shared Across All Stages, While ‘S’ Means That the Input of SR Branch in Current Stage Is the Output From the Last Stage, and the SR Branch Is Non-Shared Across Stages

N		1	2	3	4	5
ACC	L	49.8%	50.9%	51.8%	52.6%	53.1%
	S	49.8%	**51.6%**	**52.4%**	**53.1%**	**53.4%**
PSNR	L	20.97	**20.99**	**21.34**	**21.22**	**21.32**
	S	20.97	20.37	19.92	20.05	19.52
SSIM	L	0.7718	**0.7633**	**0.7774**	**0.7785**	**0.7780**
	S	0.7718	0.7622	0.7557	0.7598	0.7544

3) Impact of Multiple Stages in TPGSR: In addition to refining the TP Generator, recurrently inputting the estimated HR image into the TPGSR can also enhance the quality of TP since the SR Module can improve the estimated HR text image in each recurrence. To find out how well the multi-stage refinement can reach, we set the stage number $N = 1, 2, \ldots, 5$ and report the text recognition accuracy in Table III. We can see that the recognition accuracy increases with the increase of N; however, the margin of improvement decreases with N. When $N = 5$, the accuracy of ‘Hard’ split begins to fall. Considering the balance between the computational cost and the performance gain, we set N to 3 in our following experiments.

4) Parameter Sharing Strategy: To determine the best sharing strategies, we conduct experiments to test on the TP Module and the SR Module. As shown in Table IV, we find that under different settings of stage number, the setting of non-shared TP Module shows significant performance improvement. However, when we use non-shared SR Module, little performance improvement on text recognition is achieved. Thus we use the settings of shared SR Module and non-shared TP Module in our multi-stage model.

5) The Input of SR Branch: In the multi-stage version of our TPGSR model, the text image will be enhanced at each stage. One intuitive question is: can we take the SR output of last stage as the input to the SR branch of current stage to further enhance the SR recovery performance? To find out the answer of this question, we intake the output SR image from the last stage and resize it to fit the input size (16×64) of current stage. Since the input to different stages are different, the shared SR branch by all stages cannot work well. We therefore adopt non-shared SR branches across stages in the experiments, and denote this setting as S. The SR text recognition rates by CRNN [17] and the PSNR/SSIM indices of TPGSR outputs under setting S and the original setting (denoted by L) are listed in Table VI. One can see that the SR recognition results under setting S show improvements over setting L; however, the PSNR/SSIM indices of setting S are inferior to that of setting L. This means that setting S can improve the text character recognition performance but largely reduce the text image perceptual quality.

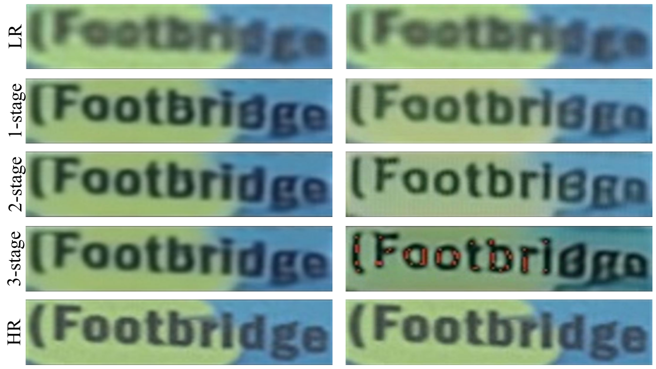
Fig. 8 visualizes the SR outputs of a text image. One can see that the SR images under setting S show clearer character shapes as the stage number increases. However, there are more artifacts and noise in the reconstructed images as well, resulting in lower image quality and hence lower PSNR and SSIM scores. In comparison, the multi-stage TPGSR outputs under setting L can maintain good visual quality while improving the SR text recognition rate. Therefore, we adopt the setting L in this paper.

6) The Effectiveness of SR in TPGSR: Since one of the goals of STISR is to improve the text recognition performance by HR image recovery, it is necessary to check if the estimated SR images truly help the final text recognition task. To this end, we evaluate the TPGSR models with both fixed and tuned TP Generator by using LR and SR images as inputs. For multi-stage version, we test all the TP Generators and pick the best LR and SR results from them. Note that models with tuned TP Generator and LR image as input is similar to directly fine-tuning the text recognition model on the LR images. The results are shown in Table V. It can be seen that by tuning TP Generator on the LR images, the text recognition accuracy can be increased. However, the recognition accuracy can be improved more by using the SR text image. For example, at stage one, the recognition accuracy

TABLE VII

SR Text Image Recognition Performance of Competing STISR Models on TextZoom. The Recognition Accuracies Are Evaluated by the Officially Released Models of ASTER [46], MORAN [47] and CRNN [17]

Approach	ASTER [46], [48]				MORAN [47], [49]				CRNN [17], [50]			
	easy	medium	hard	**average**	easy	medium	hard	**average**	easy	medium	hard	**average**
BICUBIC	64.7%	42.4%	31.2%	47.2%	60.6%	37.9%	30.8%	44.1%	36.4%	21.1%	21.1%	26.8%
SRCNN [5], [51]	69.4%	43.4%	33.0%	49.5%	63.2%	39.0%	30.2%	45.3%	38.7%	21.6%	20.9%	27.7%
TPGSR-SRCNN	72.9%	50.7%	34.7%	53.8%	67.7%	49.7%	32.8%	50.9%	47.0%	30.6%	24.7%	34.7%
SRResNet [5], [7]	69.6%	47.6%	34.3%	51.3%	60.7%	42.9%	32.6%	46.3%	39.7%	27.6%	22.7%	30.6%
TPGSR-SRResNet	76.0%	58.8%	40.1%	59.1%	72.3%	54.9%	38.4%	56.0%	54.6%	41.2%	32.3%	43.3%
RDN [5], [20]	70.0%	47.0%	34.0%	51.5%	61.7%	42.0%	31.6%	46.1%	41.6%	24.4%	23.5%	30.5%
TPGSR-RDN	72.6%	54.2%	37.2%	55.5%	67.8%	51.7%	36.0%	52.6%	53.0%	38.0%	27.7%	40.2%
TSRN [5]	75.1%	56.3%	40.1%	58.3%	70.1%	53.3%	37.9%	54.8%	52.5%	38.2%	31.4%	41.4%
TPGSR-TSRN	**78.9%**	**62.7%**	**44.5%**	**62.8%**	**74.9%**	**60.5%**	**44.1%**	**60.5%**	**63.1%**	**52.0%**	**38.6%**	**51.8%**
HR	94.2%	87.7%	76.2%	86.6%	91.2%	85.3%	74.2%	84.1%	76.4%	75.1%	64.6%	72.4%



setting 'L' setting 'S'

Fig. 8. Comparison of the multi-stage TPGSR outputs under settings 'L' and 'S'. From top row to bottom row: bicubic LR image, TPGSR outputs at stage 1, stage 2, stage 3, and the HR image. The SR outputs on the left and right columns are from setting 'L' and setting 'S', respectively.

of LR images by using tuned TP Generator is 45.3%, while the accuracy of SR images even without fine-tuning the TP Generator is 49.8%. If the tuned TP Generator is used to generate the SR text image, the text recognition performance can be further improved compared to the fixed TP Generator. Moreover, as the stage number grows, the SR text image recognition is constantly improved by both the fixed and tuned TP generators. It reveals the stability of our multi-stage refinement. However, the performance of LR input with tuned TP Generator (*i.e.*, I_L under ACC_T) degrades as the stage number grows. The reason is that the TP Generators in latter stages are tuned on better quality recovered SR images and therefore it shows poor recognition on the LR images. In conclusion, the experiments and comparisons demonstrate the effectiveness of our SR Module in improving the final SR text recognition.

C. Comparison With State-of-the-Arts

As described in Section III-B and illustrated in Fig. 4, the SR block of most existing representative SISR and STISR models can be adopted in the SR Module of our TPGSR framework, resulting in a new TPGSR model. To verify the superiority of our TPGSR framework, we select several popular SISR models, including SRCNN [51], SRResNet [7], RDN [20], and specifically-designed STISR model TSRN [5], and embed their SR blocks into our TPGSR framework. The corresponding STISR models are called TPGSR-SRCNN, TPGSR-SRResNet, TPGSR-RDN and TPGSR-TSRN, respectively. The TextZoom [5] is used to evaluate our models as well as their prototypes, while ICDAR2015 [44] and SVT [45] datasets are used to evaluate the generalization of our best model. For fair comparison, all models are trained on TextZoom dataset with the same settings.

1) Results on TextZoom: The experimental results on TextZoom are shown in Table VII. Here we present the text recognition accuracies on STISR results by using the official ASTER [46], MORAN [47] and CRNN [17] text recognition models. In Fig. 9, we visualize the SR images by the competing models with the ground-truth text labels. From Table VII and Fig. 9, we can have the following findings.

First, from Table VII we see that our TPGSR framework significantly improves the text recognition accuracy of all original SISR/STISR methods under all settings. This clearly validates the effectiveness of TP in guiding text image enhancement for recognition. Second, from Fig. 9 we can see that with TPGSR, all SR models show clear improvements in text image recovery with more readable character stroke, resulting in correct text recognition. This also explains why our TPGSR can improve significantly the text recognition accuracy, as shown in Table VII.

2) Generalization to Other Datasets: As mentioned in Section IV-A, to verify the generalization performance of our model trained on TextZoom to other datasets, we apply it to the low quality images (height ≤ 16 or recognition score ≤ 0.9) in ICDAR2015 and SVT. Overall, 563 low quality images were selected from the $2,077$ testing images in ICDAR2015, and 104 images were selected from the 647 testing images in SVT. The STISR and text image recognition experiments are then performed on the 667 low-quality images. Since TSRN [5] is specially designed for text image SR and it performs much better than other SISR models, we only employ TSRN and TPGSR-TSRN in this experiment. The ASTER and CRNN text recognizers as well as stronger baseline SEED [52] are used.

The results are shown in Table VIII. We can have the following findings. First, compared with the text recognition results using original images without SR, TSRN improves the performance when CRNN is used as text recognizer, but

TABLE VIII
TEXT RECOGNITION ACCURACY ON THE LOW-QUALITY IMAGES IN ICDAR2015/SVT DATASETS BY THE TSRN AND TPGSR-TSRN MODELS TRAINED ON THE TEXTZOOM DATASET

Dataset	ICDAR2015			SVT		
No. of images	563			104		
Approach	SEED [52]	ASTER [46]	CRNN [17]	SEED [52]	ASTER [46]	CRNN [17]
Origin	54.0%	50.8%	21.5%	60.2%	50.8%	19.2%
TSRN [5]	52.6%	48.3%	24.5%	54.3%	48.3%	23.1%
TPGSR-TSRN	**56.1%**	**52.0%**	**27.1%**	**61.1%**	**52.0%**	**29.8%**

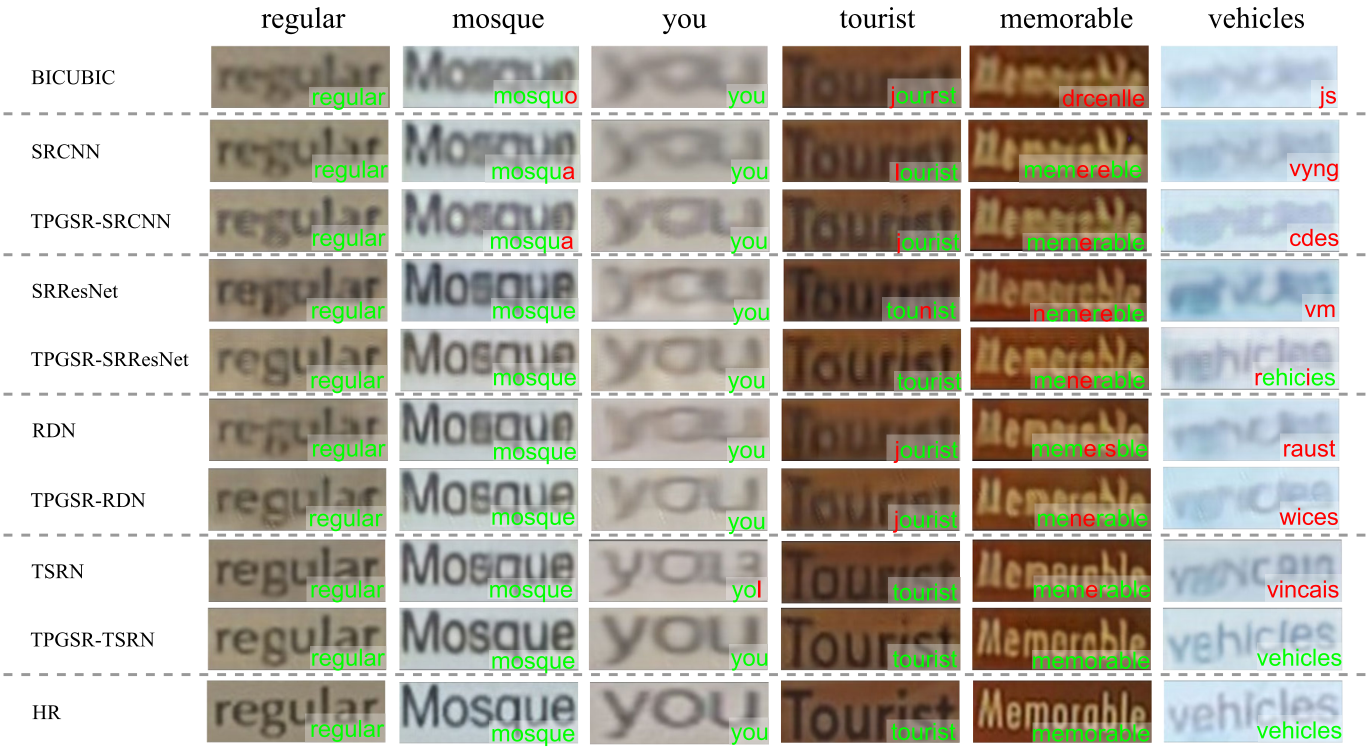


Fig. 9. Visual comparison of competing STISR models on TextZoom. The word on the bottom-right corner of each image is the text recognition result, with correct characters or words in green and wrong in red.

TABLE IX
COST *v.s.* PERFORMANCE. N REFERS TO THE NUMBER OF STAGES IN TPGSR

Super-resolver		Recognizer	
Approach	Flops	ASTER [46] (4.72G)	CRNN [17] (0.81G)
[5] w 5 SRBs	0.91G	58.3%	41.4%
[5] w 7 SRBs	1.16G	57.7%	40.1%
[5] w 9 SRBs	1.41G	57.1%	40.0%
[5] w 12 SRBs	1.78G	56.6%	39.8%
ours $(N = 1)$	1.76G	**60.9%**	**49.8%**

reduces the performance when ASTER or SEED is used as the recognizer. This implies that TSRN does not have stable cross-dataset generalization capability. Second, TPGSR-TSRN can consistently improve the performance over the original images for all the three recognizers. This demonstrates that it has good generalization performance on cross-dataset test. Third, TPGSR-TSRN consistently outperforms TSRN under all settings.

3) Cost v.s. Performance: To further examine the effectiveness of our TPGSR, we compare the computational cost of our single-stage TPGSR with TSRN [5]. In Table IX, we perform experiments of TSRN with different number of Sequential-Residual Blocks (SRBs). The results show that straightly increasing the number of SRBs is not an effective way to improve the performance of TSRN (results in accuracy drop with more SRBs). However, under our designed TPGSR network, the performance improves by 8.4% with CRNN [17] and 2.6% with ASTER [46] compared to TSRN with 5 SRBs. It is humble to conclude that exploiting text prior under our TPGSR framework deserves the additional cost it introduces. Moreover, compared to CRNN [17], ASTER [46] performs better in recognition task with higher computational cost (0.81G *v.s.* 4.72G).

D. Discussions

1) Selection of TP: There are different choices of TP Generator in our framework, *e.g.*, CTC-based generator

TABLE X

TP Types on SR Text Image Recognition. I_L, $\hat{I}_H$, I_H and 'TL' Stand for LR, SR and HR Image Input to the TP Generator and the Text Label, Respectively

TP Input	tuned	ACC
I_H	×	58.0%
I_L	×	44.5%
I_L	✓	49.8%
$\hat{I}_H$	✓	**51.8%**
TL	-	45.9%

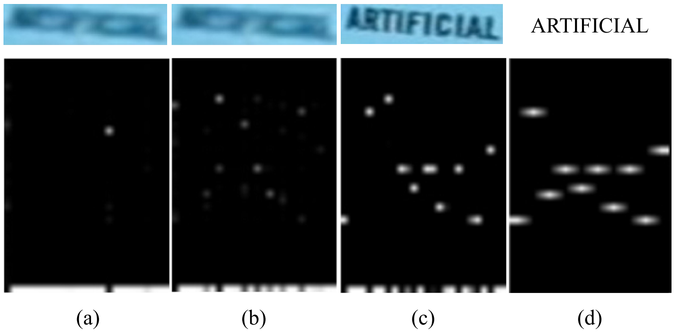


Fig. 10. Visualization of different types of TP. The first row shows the input and the second row visualizes how TP looks like. (a) to (d) are results by LR with Fixed TP Generator, LR with tuned TP Generator, SR with tuned TP Generator and one-hot text label.

TABLE XI

The SR Text Image Recognition Rate and Computational Cost by Using Different Text Recognition Models as TPG. 'ACC' Means the SR Text Recognition Rate by the CRNN [17], [50] Text Recognizer

TPG type	ACC	Model Size	Flops
CRNN [17]	49.8%	8.3M	1.7G
CRNN [17] (ResNet-26 [53])	51.1%	16.4M	4.0G
ASTER [46]	42.1%	21.0M	4.7G
ASTER [46] w random spacing	47.6%	21.0M	4.7G
Ground-truth text label	45.9%	-	-

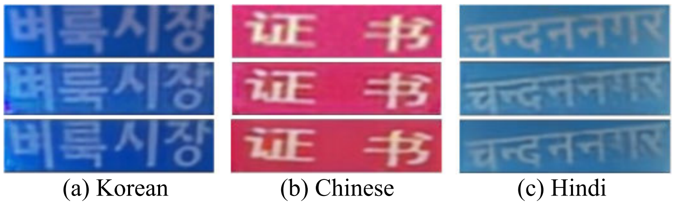

(a) Korean (b) Chinese (c) Hindi

Fig. 11. Examples of out-of-category text image SR in different languages. From top to bottom: the LR image and super-resolved HR images by TSRN [5] and our TPGSR-TSRN.

such as CRNN [17] and attention-based generator such as ASTER [46]. As introduced in III-A, TP generated from CTC-based model presents the foreground and background categorical text prediction by order as in the input image. Fig. 10 visualizes the TP generated by CTC-based model with different inputs. The lighter the points on the TP, the higher the probabilities of corresponding characters are. For the input LR text image, the tuned TP Generator can yield a clearer representation of TP. Compared with Fig. 10(a), the probabilities in Fig. 10(b) are sharper with higher categorical probability on correct characters. The SR text recognition performance using tuned TP Generator can reach 49.8%, 5.3% higher than the fixed TP Generator (44.5%) as shown in Table V. If we further input the recovered SR image to the tuned TP Generator, we can have an even better TP estimation as shown in Fig. 10(c), resulting in another 2.0% gain compared with the LR inputs (refer to Table X). However, TP estimated by the attention-based model predicts only the foreground characters. To test the upper-bound of attention-based TP, we directly use the one-hot ground truth label as the TP input (shown in Fig. 10(d)). The result in Table V illustrates inferior performance to the CTC-based TP (45.9% vs. 49.8%) and it is also far behind the upper bound of the CTC-based TP by HR input (45.9% *v.s.* 58.0%).

2) The Selection of TP Generator (TPG): Most of the commonly-used text recognition models are either CTC-based (*e.g.*, CRNN [17] in our TPG) or attention-based (*e.g.*, ASTER [46] and MORAN [47]). CTC-based TPG predicts results with both foreground and background labels, while attention-based TPG predicts only the foreground labels. In CTC-based prediction, the foreground labels indicate what the characters are, while the background labels indicate the background area. According to the spacing between two characters, the CTC-based model could employ different amount of background labels (demonstrated in Fig. 2) to indicate the background space between the neighboring characters. Such an arrangement could well align the text prior to the image feature and hence allow the proper guidance for recovering text characters.

Compared with CTC-based TPG, the attention-based TPG could not provide background label spacing. Referring to Table XI, one can see that without background label spacing, even the ground truth text label (the upperbound of attention-based text prior) could not provide effective guidance, achieving only an SR text recognition rate of 45.9%. By using ASTER [46] as TPG, we can only achieve a recognition rate of 42.1%. If we insert random blanks into the ASTER prediction as text prior, the SR text recognition rate can be improved to 47.6%, which is still much worse than the guidance from CRNN [17] (49.8%). In conclusion, the CTC-based model can generate better adaptive spacing between the foreground characters and result in better SR guidance. Moreover, using ASTER as TPG will largely increase the computational cost and model size.

On the other hand, a better CTC-based recognition model can enhance the final SR text recognition. If we upgrade the backbone of CRNN [17] from a 7-layer VGG-Net to a 26-layer ResNet [53] and use it as the TPG, the SR recognition rate can be further improved from 49.8% to 51.1%. However, the computational cost will also be increased by 2.4 times in terms of Flops. Therefore, we keep the original CRNN model as the TPG to balance accuracy and efficiency.

3) Out-of-Category Analysis: As mentioned in Section III-E and the implementation in the original repository [17], [50], the TP Generator will automatically assign the out-of-category label with blank labels in the pre-training phase. In inference, when the input is an

TABLE XII

INFERENCE TIME (FPS) AND SR TEXT RECOGNITION ACCURACY OF TPGSR-TSRN WITH DIFFERENT NUMBER OF STAGES. THE EXPERIMENTS ARE CONDUCTED WITH A SINGLE RTX 2080TI GPU

Recognizer	TPGSR (stage number)	ACC	Inference FPS
CRNN [17]	×	26.8%	730.3
	✓ (1)	49.8%	408.8
	✓ (2)	50.9%	357.3
	✓ (3)	51.8%	280.6
	✓ (4)	52.6%	201.4
	✓ (5)	53.1%	115.3
MORAN [47]	×	44.1%	91.1
	✓ (1)	54.8%	78.9
	✓ (2)	57.9%	70.1
	✓ (3)	60.5%	62.8
	✓ (4)	60.9%	54.5
	✓ (5)	61.0%	48.0
ASTER [46]	×	47.2%	31.7
	✓ (1)	60.9%	29.6
	✓ (2)	62.0%	26.6
	✓ (3)	62.8%	24.1
	✓ (4)	62.7%	20.5
	✓ (5)	62.7%	18.1

out-of-category (*e.g.*, Chinese or Korean) text image, there is a high probability that CRNN will classify it to the blank category, and thus provide null TP guidance for SR recovery. For such characters, the STISR results will mainly depend on the SR Module in our TPGSR network. To test the SR performance of our TPGSR model on images with out-of-category characters, we applied it to some text images in Korean, Chinese and Hindi picked from the ICDAR-MLT [54] dataset. The results are shown in Fig. 11. We see that the reconstructed HR text images by our model show clearer appearance and contour than their LR counterparts. This thus shows the robustness of our TPGSR in handling out-of-category text image recovery.

4) Inference Speed: As we discussed in Section IV-B.2, the increase of stage number will result in the decrease of inference speed. Here we list the inference speed of the whole processing (super-resolution and text recognition) in Table XII. The experiments are conducted with a single RTX 2080TI GPU. One can see that with the increase of stage number in our TPGSR, the recognition accuracy on TextZoom increases but the inference speed drops. Considering that the accuracy gains of 4-stage and 5-stage TPGSR over 3-stage TPGSR is not stable (*e.g.*, 0.4% and 0.5% rise for MORAN [47], and 0.1% drop for ASTER [46]), while the inference speed drops much for 5-stage setting (*e.g.*, from 62.8FPS to 48.0FPS for MORAN [47] and from 24.1FPS to 18.1FPS for ASTER [46]), we choose to use 3 stages as our default setting.

5) Failure Case: Though TPGSR can improve the visual quality of SR text images and boost the performance of text recognition, it still has some limitations, as shown in Fig. 12. First, if the text instance encounters certain perspective distortion, the benefit brought by TPGSR will be weakened. Fig. 12(a) shows such an example. Though the recognition result is correct, the improvement on image quality is not significant. Second, for the cases with extremely compressed text instance, as shown in Fig. 12(b), the TP Generator may fail with wrong recognition outputs due to the information loss in the source image. In such case, the TP feature will not effectively help the character reconstruction. In addition, the text prior refinement strategy assumes that the text prior estimated from SR text image can provide more knowledge for the TP Generator to refine text prior guidance. If the LR input can already provide good text prior, or the SR text image generated in previous stage cannot provide additional information, our refinement strategy will not provide further benefit for SR text image recovery.

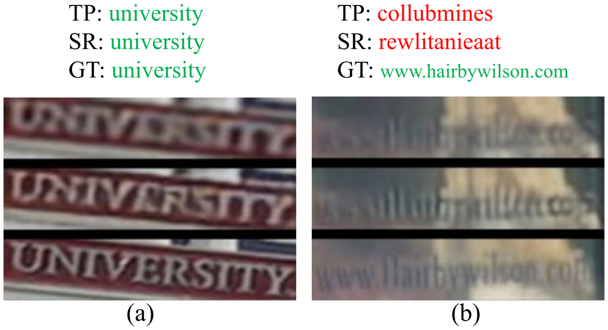


Fig. 12. Examples of failure cases. (a) Multi-oriented text. (b) Extreme compressed text.

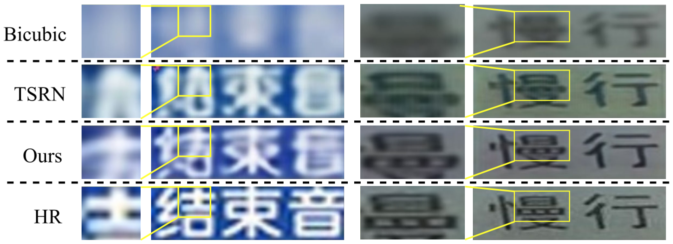


Fig. 13. Feasibility study on the Chinese text recovery.

To address the above issues, in the future we could consider adopting more powerful TP Generators to provide more robust guidance, and design new TP guidance strategies for recovering multi-oriented scene text and curve text. In addition, the failures on compressed text may be alleviated by using more powerful TP Generator that can deal with compressed text recognition.

6) Recovering Hieroglyphs (e.g., Chinese): In this work, we focused on the real-world SR of English text images, for which there is a well-prepared benchmark dataset TextZoom. It is interesting to know whether our proposed method can be adopted for hieroglyphs such as Chinese. Here we perform some preliminary experiments to validate the feasibility. We train a multilingual recognition model using CRNN on the ICPR2018-MTWI Chinese and English dataset [55] as our TP Generator. The overall alphabet contains 3, 965 characters, including the English and Chinese frequently-used set. Since there is no real-world benchmark dataset with LR-HR image pairs of Chinese characters, we synthesize LR-HR text image pairs by Gaussian blurring with blur kernel size of 5 × 5 and half-sized down-sampling the MTWI text images. We inherit the splits of MTWI as our training (59, 886 samples) and testing (4, 838 samples) set. The model training and testing are conducted following the settings described in Section III-E.

The SR text recognition results are 27.7% (Bicubic), 41.1% (TSRN [5]), 42.7% (TPGSR-TSRN) and 56.1% (HR). From the results, we can observe that our TPGSR framework can still achieve 1.6% accuracy gain over TSRN. Visualization on some Chinese characters can be seen in Fig. 13. One can see that our TPGSR can improve much the visual quality of SR results. Compared to TSRN [5], our TPGSR can better recover the text stroke on the samples. This preliminary experiment verifies that our TPGSR framework can be extended to hieroglyphs. More investigations and real-world dataset construction will be made in our future work.

V. Conclusion

In this paper, we presented a novel scene text image super-resolution framework, namely TPGSR, by introducing text prior (TP) to guide the text image super-resolution (SR) process. Considering the fact that text images have distinct text categorical information compared with those natural scene images, we integrated the TP features and image features to more effectively recover the text characters. The enhanced text image can produce better TP in return, and therefore multi-stage TPGSR was employed to progressively improve the SR recovery of text images. Experiments on TextZoom benchmark and other datasets showed that TPGSR can clearly improve the visual quality and readability of low-resolution text images, especially for those hard cases, and consequently improve significantly the text recognition performance on them.

References

[1] S. Karaoglu, R. Tao, T. Gevers, and A. W. M. Smeulders, "Words matter: Scene text for image classification and retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1063–1076, May 2016.

[2] C. Y. Fang, C. S. Fuh, P. S. Yen, S. Cherng, and S. W. Chen, "An automatic road sign recognition system based on a computational model of human recognition processing," *Comput. Vis. Image Understand.*, vol. 96, no. 2, pp. 237–268, Nov. 2004.

[3] S. Montazzolli Silva and C. Rosito Jung, "License plate detection and recognition in unconstrained scenarios," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 580–596.

[4] A. F. Biten et al., "Scene text visual question answering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4291–4301.

[5] W. Wang et al., "Scene text image super-resolution in the wild," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 650–666.

[6] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 391–407.

[7] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.

[8] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 136–144.

[9] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 286–301.

[10] H. T. M. Tran and T. Ho-Phuoc, "Deep Laplacian pyramid network for text images super-resolution," in *Proc. IEEE-RIVF Int. Conf. Comput. Commun. Technol. (RIVF)*, Mar. 2019, pp. 1–6.

[11] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 624–632.

[12] Z. Bílková and M. Hradiš, "Perceptual license plate super-resolution with CTC loss," *Electron. Imag.*, vol. 32, no. 6, p. 52, Jan. 2020.

[13] W. Wang et al., "TextSR: Content-aware text super-resolution guided by recognition," 2019, *arXiv:1909.07113*.

[14] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 369–376.

[15] X. Wang, K. Yu, C. Dong, and C. C. Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 606–615.

[16] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang, "FSRNet: End-to-end learning face super-resolution with facial priors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2492–2501.

[17] B. Shi, X. Bai, and C. Yao, "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.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2016.

[18] H. Chen et al., "Real-world single image super-resolution: A brief review," *Inf. Fusion*, vol. 79, pp. 124–145, Mar. 2022.

[19] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2015.

[20] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.

[21] X. Xu, Y. Ma, and W. Sun, "Towards real scene super-resolution with raw images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1723–1731.

[22] A. Shocher, N. Cohen, and M. Irani, "'Zero-shot' super-resolution using deep internal learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2018, pp. 3118–3126.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[24] C. Dong, X. Zhu, Y. Deng, C. C. Loy, and Y. Qiao, "Boosting optical character recognition: A super-resolution approach," 2015, *arXiv:1506.02211*.

[25] C. Peyrard, M. Baccouche, F. Mamalet, and C. Garcia, "ICDAR 2015 competition on text image super-resolution," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1201–1205.

[26] Y. Mou et al., "PlugNet: Degradation aware scene text recognition supervised by a pluggable super-resolution unit," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 158–174.

[27] X. Zhang, Q. Chen, R. Ng, and V. Koltun, "Zoom to learn, learn to zoom," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3762–3770.

[28] J. Cai, H. Zeng, H. Yong, Z. Cao, and L. Zhang, "Toward real-world single image super-resolution: A new benchmark and a new model," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3086–3095.

[29] X. Xu, D. Sun, J. Pan, Y. Zhang, H. Pfister, and M.-H. Yang, "Learning to super-resolve blurry face and text images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 251–260.

[30] Y. Quan, J. Yang, Y. Chen, Y. Xu, and H. Ji, "Collaborative deep learning for super-resolving blurry text images," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 778–790, 2020.

[31] J. Chen, B. Li, and X. Xue, "Scene text telescope: Text-focused scene image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12026–12035.

[32] C. Zhao et al., "Scene text image super-resolution via parallelly contextual attention network," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 2908–2917.

[33] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 512–528.

[34] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," 2015, *arXiv:1506.04395*.

[35] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, 2016.

[36] W. Liu, C. Chen, K.-Y. K. Wong, Z. Su, and J. Han, "STAR-Net: A spatial attention residue network for scene text recognition," in *Proc. Brit. Mach. Vis. Conf.*, vol. 2, 2016, p. 7.

[37] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5076–5084.

[38] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "AON: Towards arbitrarily-oriented text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2018, pp. 5571–5579.

[39] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[40] K. Chen et al., "Hybrid task cascade for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4974–4983.

[41] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2315–2324.

[42] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," 2014, *arXiv:1406.2227*.

[43] Q. Xie, M. Zhou, Q. Zhao, D. Meng, W. Zuo, and Z. Xu, "Multispectral and hyperspectral image fusion by MS/HS fusion net," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1585–1594.

[44] D. Karatzas et al., "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1156–1160.

[45] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1457–1464.

[46] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, Sep. 2018.

[47] C. Luo, L. Jin, and Z. Sun, "MORAN: A multi-object rectified attention network for scene text recognition," *Pattern Recognit.*, vol. 90, pp. 109–118, Jun. 2019.

[48] *ASTER in Pytorch*. Accessed: Sep. 12, 2021. [Online]. Available: https://github.com/ayumiymk/aster.pytorch

[49] *MORAN: A Multi-Object Rectified Attention Network for Scene Text Recognition*. Accessed: Sep. 12, 2021. [Online]. Available: https://github.com/Canjie-Luo/MORAN_v2

[50] *Convolutional Recurrent Network in Pytorch*. Accessed: Sep. 12, 2021. [Online]. Available: https://github.com/meijieru/crnn.pytorch

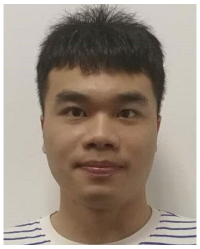
[51] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 184–199.

[52] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "SEED: Semantics enhanced encoder–decoder framework for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13528–13537.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

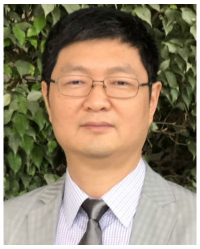
[54] N. Nayef et al., "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification-RRC-MLT," in *Proc. ICDAR*, vol. 1, 2017, pp. 1454–1459.

[55] *ICPR 2018 Contest on Robust Reading for Multi-Type Web Images (MTWI)*. Accessed: Mar. 22, 2019. [Online]. Available: https://tianchi.aliyun.com/getStart/introduction.htm?spm=5176.100066.0.0.50c233aftaCagb&raceId=231686



Jianqi Ma received the B.Sc. and M.Sc. degrees from Fudan University in 2015 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University. His research interests include text image enhancement, text image understanding, and image processing pipeline.

Shi Guo received the B.Sc. and M.Sc. degrees from the Harbin Institute of Technology in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University. His research interests include image/video enhancement, image/video denoising, and image processing pipeline.



Lei Zhang (Fellow, IEEE) received the B.Sc. degree from the Shenyang Institute of Aeronautical Engineering, Shenyang, China, in 1995, and the M.Sc. and Ph.D. degrees in control theory and engineering from Northwestern Polytechnical University, Xi'an, China, in 1998 and 2001, respectively. In 2006, he joined the Department of Computing, The Hong Kong Polytechnic University, as an Assistant Professor. Since July 2017, he has been a Chair Professor with the Department of Computing, The Hong Kong Polytechnic University. His research interests include computer vision, image and video analysis, pattern recognition, and biometrics. He has published more than 200 papers in those areas. As of 2022, his publications have been cited more than 81,000 times in literature. He is a Senior Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and is/was an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *SIAM Journal on Imaging Sciences*, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is a "Clarivate Analytics Highly Cited Researcher" from 2015 to 2022. More information can be found at: http://www4.comp.polyu.edu.hk/∼cslzhang/