Detecting and Removing Text in the Wild

JUNHO CHO , (Member, IEEE), SANGDOO YUN, DONGYOON HAN, BYEONGHO HEO, AND JIN YOUNG CHOI , (Member, IEEE)

1Automation and Systems Research Institute, Seoul 08826, South Korea
2Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
3NAVER AI LAB, Seongnam-si, Gyeonggi-do 13494, South Korea

Corresponding author: Jin Young Choi (jychoi@snu.ac.kr)

This research was supported by the IITP (Institute for Information & Communication Technology Promotion) grant funded by the MSIT (Ministry of Science and ICT, Korea) as [2017-0-00306, Outdoor Surveillance Robots; No. B0101-15-0266, Visual Big-Data Discovery Platform] with the help of NSML (Naver Smart Machine Learning Platform) while Junho Cho was a research intern at NAVER CLOVA AI Research Team.

ABSTRACT Scene text removal is a challenging task that aims to erase wild text regions that include text strokes and their ambiguous boundaries, such as embossing, shade, or flare. The challenging issues raised in the wild are not completely addressed by the existing methods. To address these issues, we propose a new loss function for blending two tasks in a new network structure that depicts wild text regions in a soft mask and selectively inpaints them into a sensible background. The proposed loss function aids the learning of two seemingly separate tasks in a synergistic way via the soft mask to achieve remarkable performance in scene text removal. We validate our method through qualitative and quantitative comparisons, and region-wise analysis, showing that our method outperforms existing methods.

INDEX TERMS Machine learning, machine vision, scene text removal.

I. INTRODUCTION

Scene text removal is the total removal of text from an image as if there were no text originally. Thus, it aims to erase text and restore a proper background. Scene text removal has been widely used to secure sensitive private information, such as names and addresses, and utilized for other computer vision tasks, such as text replacement for language swaps. Unlike image inpainting [2]–[6], in which an explicit mask of an inpainting region is given, scene text removal requires identifying the inpainting region, capturing the location of text in an image is necessary in scene text removal. However, text in the wild consists of various types of fonts, which may be irregularly shaped and have varying levels of illumination as well as perspective distortion, making them challenging to localize. In this paper, such a text has been referred to as wild text.

Early text removal researches [9]–[13] utilized two separate techniques: text stroke segmentation [14]–[16] and image inpainting [2], [3], [17]–[19]. However, these are commonly used for removing subtitles or captions, which are printed in a straight direction on a simple background. Recent deep-learning-based methods [1], [20] have adopted a deep neural network that erases wild text in a data-driven way without explicit text localization. These methods include gathering pseudo-background data as the ground truth by manually erasing the original text in an image and trained their models using an image-regression-based inpainting loss.

FIGURE 1. Typical issues in scene text removal. The rows present (a) input images, (b) baseline [1] results, and (c) our results. Each case represents three typical failures in (b), 1. un-erased text regions, 2. falsely erased non-text regions and, 3. more complicated text un-erased in the wild.
more effectively than the early approaches [9]–[13]. However, the text removal methods still result in failure cases: falsely un-erased text regions and erased non-text regions, as shown in Figure 1 (cases 1 and 2). Furthermore, wild text can be more complicated, as seen in Figure 1 (case 3), where ambiguous text boundaries may appear around the text stroke, such as three dimensional embossing and surrounding shade. These deep-learning-based methods [1], [20] also fail to localize such challenging wild text.

The objective of our paper is to tackle the challenges and mitigate the ambiguous text boundary problems in localizing and inpainting the wild text regions. To this end, instead of utilizing an existing single net structure, we devise a two-task blending structure that performs wild text segmentation and selective background inpainting at the same time. In addition, we propose a blending loss function to train the two-task structure in an end-to-end synergistic way. With the blending structure and loss function, our model can localize ambiguous text boundaries inclusively in a soft mask and erase the wild texts completely.

In the proposed loss function, the part for wild text segmentation is designed to produce a soft mask that indicates wild text regions. The part for selective background inpainting is designed to restore a visually plausible background only in the region where the wild text is erased. The soft mask plays a key role in linking two loss functions for end-to-end training and blends the input image and the predicted background to produce a text-removed output. Through the experiments, we show our method outperforms the state-of-the-art methods [1], [20] in terms of the image quality and degradation of text detection [21], [22].

Our contributions are summarized as follows:

- We propose a two-task blending structure that performs both segmentation of a wild text area and restoration of a plausible background in that area.
- We design an end-to-end loss function to train the two-task blending structure in a synergistic way.
- The proposed model effectively handles ambiguous text boundaries, such as shade, embossing, and flare.

II. RELATED WORK

Early text-removal approaches [9]–[13] focused on erasing subtitles and captions in digital-born images. They used edge-based [16], connected component analysis-based [15] or stroke-filter-based [14] methods to segment the text stroke. However, the stroke segmentation algorithms failed to localize wild text in real scenes. Eventually, cascaded image inpainting algorithms [2], [3], [17]–[19] were able to restore the background only in text strokes. Since these approaches naively combine the independent text stroke segmentation and image inpainting algorithms, they fail to remove text in wild scenes, which is the ultimate goal.

To overcome the problem of each function operating separately, deep-learning-based methods emerged to train scene text removal in a data-driven way. Scene-Text-Eraser [20] made the first attempt to erase text from real scenes using a deep neural network. To solve the data-deficiency problem of scene text removal, that is, the fact that non-text versions of real data do not exist, Scene-Text-Eraser generated 229 pseudo-background images as the ground truth from the ICDAR13 [7] dataset by masking the original images with the given stroke annotations and inpainting with the third-party algorithm [2]. Then, Scene-Text-Eraser used pairwise input images and pseudo-background images to train a deep neural network, which produced text-removed images. However, the scene text removal performance heavily relied on the heuristics of the pseudo-background generation, which followed aforementioned naive approaches [9]–[13]. Although Scene-Text-Eraser dilated the given text stroke mask to cover ambiguous text boundaries in the pseudo-background generation, it still could not deal with more complex wild text. Furthermore, the 229 generated samples were not enough to train a deep network.

The data-deficiency problem of scene text removal mainly arises because there is no dataset that annotates wild text. Therefore, EnsNet [1] simply used human labor to remove text from images. EnsNet generated 1,000 pseudo-background images by manually erasing text from the ICDAR17-MLT [23] dataset. However, the 1,000 image pairs were still not enough to train a deep network, so they obtained an additional 8,000 pairwise data by synthesizing text on random background images using the SynthText [24] engine. They trained a deep neural network to regress an output to the pairwise background ground truth in a similar way as with Scene-Text-Eraser [20]. Therefore, wild text is implicitly localized and replaced with a plausible background. Through the ResNet [25]-based framework and a multi-scale regression loss, EnsNet achieved reasonable scene text removal performance. However, EnsNet falsely removed non-text regions or leaves text in complicated scenes due to the poor text localization performance, as shown in Figures 1 and 2. EnsNet may have focused more on how to design a regression-based inpainting loss function to generate a plausible background and less on how to localize text in an image. Furthermore, the data-deficiency problem remained, as their pseudo-background generation is inefficient when a model needs to generalize to different domains of text. For example, EnsNet [1] trained its model on linearly oriented alphanumeric text, but to generalize the method for use on multi-lingual and curved text, additional costly annotation efforts are required to generate image pairs with and without such text. Other notable and recent scene text removal works are [26], [27] with additional text segmentation and GAN [28]-based inpainting approach. These methods also achieve scene text removal in good quality, but a quantitative comparison is limited as targeting dataset is different.
In summary, the previous methods [1], [20] have focused on generating pseudo-background images as the ground truth and using a sole inpainting loss to remove wild text without explicit text localization. Unlike previous methods, we do not incur high annotation costs on pseudo-background generation but rather use a synthetic text dataset with cost-free background images and a real scene-text dataset [7], [8] with text annotations. To tackle the poor text localization performance of previous methods [1], [20], we devise a model that explicitly localizes challenging wild text and includes ambiguous text boundaries in a wide variety of real scenes. Therefore, we introduce a novel architectural solution with a blending loss, which jointly trains wild text segmentation and selective background inpainting.

### III. PROPOSED METHOD

The existing methods EnsNet [1] and Scene-Text-Eraser [20] adopt a U-net [29] structure for scene text removal, which is referred to as single-net models in this paper (Figure 3 (a)) for convenience. Single-net models are trained with a sole inpainting loss function, which regresses the output to background ground truth (GT). In contrast to the single-net models, as shown in Figure 3 (b), we propose a blending structure of wild text segmentation and selective background inpainting to enhance localization and restore a plausible background in wild text.

#### A. NETWORK STRUCTURE

Our model has two modules for wild text segmentation and selective background inpainting. Both modules share features and receive skip connections [29] from the backbone network. Let $I_{in}$ be the given input image containing text.

From the backbone feature of $I_{in}$, the wild text segmentation module predicts a soft mask $\hat{M}$, which indicates regions to be removed with continuous values between 0 and 1. The selective background inpainting module predicts a background image $\hat{I}_{bg}$ to be filled in removed regions. The distinguishing aspect of this work is that the blending module constructs the text-removed image $\hat{I}_{out}$ by blending $I_{in}$, $\hat{M}$, and $\hat{I}_{bg}$ as

$$\hat{I}_{out} = (\hat{I}_{bg} \otimes \hat{M}) + (I_{in} \otimes (1 - \hat{M})), \quad (1)$$

where $\otimes$ denotes the element-wise multiplication. As the blending operation is differentiable, the back-propagated gradients from $\hat{I}_{out}$ flow into the segmentation and inpainting modules and the backbone network.

Such blending approach has been effective in other object removal or editing tasks [30], [31]. Pumarola et al. [31]
Figure 4. Difference between separate training and joint training. The input image has wild text transcribed as “SOPRA”, which contains a glowing flare around it. For the input image, text localization and text removal results are presented for when the segmentation module and the inpainting module of the two-task blending structure are (a) separately trained and (b) jointly trained.

proposed face editing framework which produces attention and color masks to compose final face-edited output from the input face image. This framework does not locate faces as its input is a full face image. The face-editing location such as eyes and lips is not available as ground truth, thus computed by the network via attention mechanism. However, wild text segmentation requires accurate text detection and ambiguous text region inclusion. While text location is annotated as strokes or boxes, ambiguous text regions such as embossing, surrounding shade, and glowing flare do not have ground truth annotation. Therefore, we divide the mask $\hat{M}$ into clear and ambiguous regions to cope with both situations when ground truth is available and not available. We will elaborate more about wild text segmentation in the following section.

1) WILD TEXT SEGMENTATION

Figure 4 illustrates a typical case of wild text, transcribed as “SOPRA” with surrounding glowing flare. We may think of simply predicting the text foreground mask $\hat{M}$ to be the text stroke mask $M$ with a binary cross entropy loss and inpainting the localized text area. Letting $M$ and $\hat{M}$ be the stroke GT and its prediction respectively, the binary cross entropy loss $L_s$ for stroke segmentation is given by

$$L_s = -\frac{1}{N} \sum_p \left( M_p \log \hat{M}_p + (1 - M_p) \log(1 - \hat{M}_p) \right)$$

where $p$ indicates the pixel position, and $N$ is the number of pixels in a mask map. As the segmentation module with binary cross entropy loss $L_s$ solely predicts the text area as text stroke and the inpainting module inpaints the localized region of $\hat{M}$ with an inpainting loss, the two modules are trained separately, and we refer to this as ‘Separate training’ in Figure 4 (a). Although separate training predicts a reasonably fine text stroke in $\hat{M}$, it fails to remove ambiguous text boundaries which remain in the output $\hat{I}_{out}$. Thus, the word ‘SOPRA’ is still readable in the text-removed output $\hat{I}_{out}$.

Accordingly, the blending loss function is proposed to jointly train both segmentation and inpainting modules for wild text segmentation. As shown in Figure 4 (b), joint training enables localizing ambiguous text boundaries inclusively in $\hat{M}$, and wild text is completely removed in $\hat{I}_{out}$. Details of the loss function will be covered in the later section.

2) SELECTIVE BACKGROUND INPAINTING

The single-net models [1], [20] directly output the text-removed image as $\hat{I}_{out}$ (see Figure 3 (a)) trained with a sole inpainting loss function, which regresses the output to background GT. That is, the single-net models unnecessarily reconstruct non-text regions in the input along with background restoration in text.

On the other hand, our selective background inpainting outputs $\hat{I}_{bg}$ (see Figure 3 (b)), in which text regions from $\hat{M}$ are selectively inpainted with background restoration. Therefore, as there is no need for redundant input reconstruction, non-text regions are bypassed from the input image.

B. TRAINING SCHEME

1) PRETRAINING WITH TEXT STROKE SEGMENTATION TASK

We pretrain the backbone network and the wild text segmentation module with stroke segmentation task as the prior knowledge for scene text removal. For the pretraining, we use the stroke segmentation loss $L_s$ defined in Eq. (2).

2) BLENDING LOSS FOR END-TO-END TRAINING

After the pretraining phase, we train the two-task blending structure by minimizing the following end-to-end objective with the aforementioned two-task outputs:

$$L = \lambda_m L_m(\hat{M}) + \lambda_i L_i(\hat{M}, \hat{I}_{bg}),$$

where $L_m$ is the loss function for explicit text stroke segmentation, and $L_i$ is the loss function for ambiguous text boundaries inclusion and selective background inpainting. Because $\hat{M}$ is included in both of the loss terms, our models can be trained so that $\hat{M}$ produces a soft mask including ambiguous text boundaries. We use the weighting parameters of $\lambda_m$ and $\lambda_i$ for balancing the corresponding losses. Through the end-to-end learning, the wild text segmentation module is trained to produce a soft mask $\hat{M}$, which can then guide the selective background inpainting module, producing an appropriate $\hat{I}_{bg}$.

That is, the wild text segmentation module receives two ways of supervision from $L_m(\hat{M})$ and $L_i(\hat{M}, \hat{I}_{bg})$, and this is the major difference from the other methods [1], [20] whose models are trained by a sole image inpainting loss. As shown in Figure 5, additional information ($M, \hat{M}$, and $I_{bg}$) is needed to construct the loss, but it is omitted in Eq. (3) for simplicity.

The loss functions $L_m$ and $L_i$ are applied to different regions. As illustrated in Figure 6, we distinguish the masks $M$, $\hat{M}$, and the region $A$. $\hat{A}$. $M$ or $\hat{M}$ denotes the ground truth of a binary mask for text stroke or text box covering the text stroke, respectively. The positive pixels in $M$ and $\hat{M}$ are encoded by one, and others by zero. Then the text box
in $\hat{M}$ is annotated so that $\hat{M} \otimes (1 - \hat{M}) = 0$, i.e., text stroke must be inside the text box. The region $A$ is defined by a set of pixels which belong to text stroke or clear background. That is, $A = \{p|M_p = 1 \lor \hat{M}_p = 0\}$ where $p$ is a pixel position and $\lor$ is logical OR operation. Then $A^c$ becomes an ambiguous text region that includes pixels outside the text stroke and inside the text box. The reason to define $M, \hat{M},$ and $A/A^c$ is to handle ambiguous text regions (shade, embossing, flare, etc.), which is a challenging task in our paper.

By using $A$, the loss $L_m$ in Eq. (4) takes a role of learning the mask $\hat{M}$ for region $A$ (text stroke and clear background region except the ambiguous region). Because there is no ground truth that annotates the ambiguous text region, we cannot include the learning the mask to predict $\hat{M}$ for the region $A^c$ by $L_m$ in Eq. (4). For the remaining area $A^c$ (ambiguous text region) for the predicted mask $\hat{M}$, we further designed the loss $L_i$ in Eq. (5).

$L_m$ (Explicit Text Stroke Segmentation): There is no such ground truth that annotates the ambiguous text region, thus $L_m$ is designed not to penalize pixels in region $A^c$. We use the binary cross entropy on each pixel of $A$ as the loss, i.e.,

$$L_m = -\frac{1}{N_A} \sum_{p \in A} \left(M_p \log \hat{M}_p + (1 - \hat{M}_p) \log(1 - \hat{M}_p)\right) \quad (4)$$

where $N_A$ is the number of valid pixels in $A$ for normalization. Note that the second term in Eq. (2) is to learn non-text regions as negative samples by supervising with $(1 - M_p)$. In Eq. (4), $(1 - \hat{M}_p)$ is used instead of $(1 - M_p)$ of Eq. (2) for excluding the ambiguous area $A^c$. The purpose of Eq (4) $L_m$ is to binary classify pixels in region $A$ of the predicted mask $\hat{M}$ so that the pixels in the text stroke follow $M_p = 1$ and the pixels in the backgrounds follow $\hat{M}_p = 0$, i.e., $1 - \hat{M}_p = 1$.

For the remaining area $A^c$ in the mask $\hat{M}$, we design $L_i$.

$L_i$ (Selective Inpainting & Text Boundaries Inclusion):

While single-net methods (Figure 3 (a)) restore background of an entire image region, our proposed structure (b) and the loss function $L_i[I_{out}(\hat{M}, \hat{I}_{bg})]$ train our model to predict a mask $\hat{M}$ that indicates where edit is needed and a background image $\hat{I}_{bg}$. With blended output $\hat{I}_{out}$ from $\hat{M}, \hat{I}_{bg}$ and Eq. (1), the loss function $L_i$ induces the background $\hat{I}_{bg}$ inpainting selectively on the positive regions of $\hat{M}$. We utilize text box GT $\hat{M}$ to mask back-propagated gradients by the loss $L_i$ to train background restoration only in $\hat{M}$. To remove text textures completely, not only text stroke but also ambiguous text boundaries should be covered up in the predicted mask $\hat{M}$. Since there is no explicit ground truth annotation for ambiguous text region, the text-free background ground truth is utilized to train our model to output non-text texture background. Therefore, the ambiguous text region in area $A^c$ is included in the mask $\hat{M}$ for complete text removal in $\hat{I}_{bg}$. $L_i$ is designed with the $L_1$ loss $L_{l1}$, the style loss $L_{stl}$ [32], the perceptual loss $L_{perc}$ [33], and the total variation loss $L_{tv}$ [33]:

$$L_i = \lambda_{l1} L_{l1} + \lambda_{perc} L_{perc} + \lambda_{stl} L_{stl} + \lambda_{tv} L_{tv}. \quad (5)$$

The $L_1$ loss $L_{l1}$ minimizes pixel distance between the output $\hat{I}_{out}$ and ground truth $\hat{I}_{bg}$ in the text box regions $M$ as

$$L_{l1} = ||\hat{M} \otimes (\hat{I}_{out} - \hat{I}_{bg})||_1. \quad (6)$$

Because the $L_1$ loss could result in blurry output, the perceptual loss $L_{perc}$ is adopted for contextual image restoration and the style loss $L_{stl}$ is adopted for better texture restoration. $L_{perc}$ and $L_{stl}$ use the ImageNet [34] pretrained VGG16 [35] model as the feature extractor $\phi(\cdot)$ and compute feature distance between the output and the background GT. The perceptual loss $L_{perc}$ is defined by

$$L_{perc} = ||\phi(\hat{I}_{out}) - \phi(\hat{I}_{bg})||_1, \quad (7)$$

which drives $\hat{I}_{out}$ to have similar semantic features to $\hat{I}_{bg}$. The style loss $L_{stl}$ is defined by

$$L_{stl} = ||\phi(\hat{I}_{out})^T \phi(\hat{I}_{out}) - \phi(\hat{I}_{bg})^T \phi(\hat{I}_{bg})||_1/N_D, \quad (8)$$

where $N_D$ denotes the size of the feature map. The style loss causes $\hat{I}_{out}$ to have a similar style as $\hat{I}_{bg}$. We extracted feature maps at the {1, 2, 3}-th pooling layers of the VGG16 model.

**FIGURE 5.** Blending loss. Schematic illustration of the end-to-end loss designed for joint training wild text segmentation and selective background inpainting. Note that only $M$ and $I_{bg}$ are our model predictions.

**FIGURE 6.** Visualization of masks and regions. (a) $M$: the text stroke mask, (b) $\hat{M}$: the text box mask, (c) $A$: the region defined by the set of pixels which are either text or background ($A = \{p|M_p = 1 \lor \hat{M}_p = 0\}$) and the loss $L_m$ is applied on $A$ of the predicted soft mask $\hat{M}$ to binary classify each pixel, (d) $A^c$: the region where the loss $L_i$ is applied to include ambiguous text-boundaries in the predicted soft mask $\hat{M}$.\]
for $L_{perc}$ and $L_{stl}$. The total variation loss $L_{tv}$ encourages spatial smoothness by minimizing the distance among adjacent pixels $(x, y)$ of the output image $I_{out}$ as

$$L_{tv} = \sum_{(x, y)} |I_{out}(x, y) - I_{out}(x, y - 1)| + |I_{out}(x, y) - I_{out}(x - 1, y)|.$$  \hspace{1cm} (9)

C. IMPLEMENTATION DETAILS

ResNet34 [25] has been used for the backbone network of both our model and the re-implemented EnsNet [1] for a fair comparison. We have adopted four skip connections from the backbone network to the decoding features of the wild text segmentation and selective background inpainting modules. We have used bilinear up-sampling and subsequent convolutional layers with kernel = 3, pad = 1, and stride = 1 to upscale features in the segmentation module.

For the selective background inpainting module, we have used gated convolution [5], which is well known to be effective for inpainting irregularly shaped regions. Wild text in an image is usually irregularly shaped and thus the gated convolution is much more effective than the standard convolution for this task. The gated convolution is composed of two convolution operations $f_{gate}$ and $f_{feat}$ that output the gating weights and the features with the same dimension, respectively. The gated convolution is defined as

$$C(x) = \sigma(f_{gate}(x)) \otimes \phi(f_{feat}(x)), \hspace{1cm} (10)$$

where $C$ is the gated convolution operation, $\phi$ is an activation function, $\sigma$ is a sigmoid function, and $x$ is an input. The exponential linear units [37] are used as activation functions. We have used transposed convolutional layers with kernel = 4, pad = 1, stride = 2 for upsampling.

While ResNet34 uses batch normalization (BN) [38] for feature normalization, instance normalization (IN) [39] is known to be powerful in image-to-image translation tasks, such as style transfer [40], domain transfer [41], and image inpainting [4]. Similarly, we have observed that IN outperforms BN in scene text removal models. Therefore, we have used IN instead of BN after every convolutional layer in every model. In gated convolution blocks, IN is applied after element-wise multiplications. Both weighting values $\lambda_m$ and $\lambda_i$ are empirically set to 1. $\lambda_L$, $\lambda_{stl}$, $\lambda_{perc}$, and $\lambda_{tv}$ are set to 6, 120, 0.05, and 0.1, respectively, in the same way as in [6].

While single-net models only rely on pairwise data of images with text and their pseudo-background, the proposed model, which is a two-headed architectural solution with the blending loss, additionally utilizes real-world dataset with wild text. Half of the training batch for the proposed model is trained with $L_m$ on the ICDAR13 [7] and Total Text datasets [8] to localize challenging wild text in real scenes. And other half of the training batch is pairwise data from our synthetic data with the complete blending loss. Single-net models use the full batch of pairwise data. All of the implemented models are trained with a batch size of 16 and an image resolution of $512 \times 512$ with random crops and scale augmentation with a ratio from 0.5 to 3.

IV. EXPERIMENTS

In this section, we describe the datasets, evaluation metrics, and experimental results.

A. DATASETS

The ICDAR13 [7] dataset is one of the most popular scene-text detection benchmarks. The dataset has 229 training images and 233 test images, including text stroke and text box annotations.

The Total Text [8] dataset is a scene-text dataset for dealing with various orientations of text. It contains 1,255 training images and 500 test images. Total Text contains text strokes, text-bounding polygons, and text box annotations.

The SynthText++ dataset is a synthetic scene-text dataset we generated to provide richer annotations from SynthText [24]. SynthText is widely used in text detection [21], [22] and text recognition [42]. We generated the synthetic images with text upon text-free images and split them into 93,888 training images and 483 test images without sharing background images. We modified the SynthText engine to additionally produce text stroke annotations. SynthText++ provides scene-text images with text stroke, text box, and background GT.

The EnsNetSet dataset is a dataset first shown in EnsNet [1], which includes 8,000 training images and 800 test images. Since text stroke and text box annotations had not been released, we used this dataset only to evaluate and compare the scene text removal models.

The SynthText++ dataset was used to train single-net models. The ICDAR13, Total Text, and SynthText++ datasets were used to train our models. The test portion of all datasets was used for the evaluation of all models.

B. EVALUATION METRICS

We used two metrics for evaluation based on the previous works [1], [20]. First, we used the text detection methods on text-removed images and original images from the ICDAR13 and Total Text datasets. We measured the performance degradation of the detectors to estimate the degree of text erasing. Two popular text detectors (EAST [21] and CRAFT [22]) were used on the ICDAR13 dataset. CRAFT was used in the Total text dataset because it detects text with irregular shapes in the form of polygons.

Second, to measure the image quality of text-removed images, Mean Squared Error (MSE) and Structural Similarity (SSIM) [43] were used. MSE measures the mean squared error between two images, with the lower MSE values indicating better text removal performance. SSIM measures the mean structural similarity index between the two images, with the higher SSIM values indicating better text removal performance.

C. EXPERIMENTAL RESULTS

In Table 1, we report the performance degradation of text detection on text-removed images. Lower detection
performance means better scene text removal performance. Thus, this measure represents how well text is erased by each scene text removal method. The experiment was conducted using the ICDAR13 and Total Text datasets with two text detectors: EAST [21] and CRAFT [22]. We first evaluated text detection performance on original images with text. Then, we evaluated text detection performance on text-removed images with each scene text removal method and measured the degree of performance degradation. Notably, recall is an important metric for measuring the amount of detectable text remaining in text-removed outputs.
TABLE 1. Detection performance of text-removed images. Text detection models, EAST [21] and CRAFT [22], are evaluated on the text-removed results of text removal models (Scene-Text-Eraser [20], EnsNet [1], and ours) using the ICDAR13 [7] and Total Text [8] datasets. The CRAFT performance of EnsNet is evaluated from our re-implemented EnsNet. Lower text detection performance means better text removal performance. ‘Ours (separate training)’ denotes when text stroke segmentation and background inpainting tasks are trained separately. ‘Ours’ is when two tasks are jointly trained with the blending loss. Figure 4 shows how joint training effectively removes wild text. See section IV-C2 for further discussion.

| Method            | ICDAR13 [7] | Total Text [8] |
|-------------------|-------------|----------------|
|                   | Recall   | Precision | H-mean | Recall   | Precision | H-mean |
| Input Images       | 79.58     | 86.79     | 83.03   | 90.56    | 96.10     | 93.23   |
| Pix2Pix [36]       | 10.19     | 69.45     | 17.78   | -        | -         | -       |
| Scene-Text-Eraser [20] | 10.08   | 39.09     | 16.03   | -        | -         | -       |
| EnsNet [1]         | 5.66      | 73.42     | 10.51   | 9.41     | 36.59     | 14.97   |
| Single-net         | 9.77      | 37.81     | 15.53   | 12.18    | 38.28     | 18.48   |
| Ours (separate training) | 26.59  | 81.55     | 29.07   | 31.69    | 36.86     | 34.08   |
| Ours               | 4.47      | 32.03     | 7.85    | 6.30     | 28.77     | 10.34   |

TABLE 2. Measurement based on image quality. We measured the MSE, and SSIM scores of the scene text removal models using the EnsNetSet dataset. * are reported from original EnsNet [1] paper. EnsNet* shows the reproduced results from our re-implementation.

| Method            | MSE (10^-3) | SSIM (%) |
|-------------------|-------------|----------|
| Input images      | 6.96        | 91.36    |
| Scene-Text-Eraser [20] | -     | 46.13    |
| Pix2Pix [36]      | -           | 89.86    |
| EnsNet [1]        | -           | 96.44    |
| EnsNet*           | 0.718       | 95.85    |
| Single-net        | 0.818       | 94.51    |
| Ours              | 0.466       | 97.06    |

1) COMPARISONS TO SINGLE-NET MODELS

In Table 1, ‘Input Images’ represents the original performance of the text detectors. ‘Pix2Pix’ [36], ‘Scene-Text-Eraser’ [20], and ‘EnsNet’ [1] are other scene text removal methods. ‘Single-net’ is the single-net structure shown in Figure 3 (a), which is trained with a sole inpainting loss function. ‘Ours’ is the proposed two-task blending structure with the novel end-to-end blending loss. ‘Ours’ outperformed every single-net model, including Scene-Text-Eraser [20], Pix2Pix [36], EnsNet [1], and our baseline ‘Single-net’ with both datasets with both detectors. The results verify that our method has superior scene text removal performance compared to the state-of-the-art methods. To train a single-net model on scene-text data, one should manually erase all text in the images to generate the pseudo-background GT. The manual text-erasing process with photo-editing tools such as photoshop require heavy and professional human labor, thus EnsNet is trained on very few data of 1,000 images from ICDAR17 text detection dataset and results in poor text localization performance. Meanwhile, our method with \( L_m (4) \) can easily utilize the wide variety of real data with text stroke and text box annotations, which are popular and already available annotations for text detection. Therefore, our method is more scalable and easier to generalize to different domains of text than single-net models.

Table 2 shows the scene text removal performance with the EnsNetSet dataset [1]. Two metrics (MSE and SSIM) were used to measure the image quality of text-removed images. We re-implemented EnsNet trained on SynthText++, denoted as EnsNet*, for fair comparison with

2) THE SUPERIORITY OF JOINT TRAINING OVER SEPARATE TRAINING

To demonstrate the effectiveness of the joint training scheme, we introduce another variant ‘Ours (separate training)’ (see Figure 4 (a) and Table 1). ‘Ours (separate training)’ has an identical network structure to ‘Ours’, but its segmentation and inpainting modules are trained separately by replacing \( L_m (4) \) of ‘Ours’ to \( L_s (2) \) and blocking the back-propagation path from \( I_{out} \) to \( M \). As shown in Figure 4 (a), ‘Ours (separate training)’ does not erase ambiguous text boundaries, leaving clearly legible text in the output. Therefore, ‘Ours (separate training)” demonstrates the worst performance by leaving the most detectable text in Table 1.

3) QUALITATIVE COMPARISONS

For qualitative comparisons, we display our results on challenging scene-text images in the ICDAR13 and Total Text
datasets in Figure 7. Our method was compared with the state-of-the-art method EnsNet [1]. The result in Figure 7 shows that our model outperforms EnsNet on challenging real scenes with wild text by training on scene-text data with text location annotations with an explicit text stroke segmentation loss function $L_m$ (4).

In the next section, we perform region-wise analysis to verify whether our method outperforms scene text removal models based on the single-net approach.

D. REGION-WISE ANALYSIS

In order to detect damages to non-text regions, it is necessary to divide the regions for measuring image quality. Denoting $Q$ as an image quality metric (either MSE or SSIM), the EnsNet [1] paper evaluated the image quality only in the full-image ($Q(\hat{I}_{out}, I_{bg})$), as shown in Table 2. Text usually occupies a small portion of an image, so the performance largely depends on the non-text area that occupies most of the image. Using the denoted regions in Figure 8, we further measured and reported image quality in two more regions, the inner and outer text box regions of $\tilde{M}$. The image quality evaluations of the additional regions are defined as:

Text regions: $\frac{1}{n} \sum_i Q(\hat{I}_{out, i}, I_{bg, i}),$ (11)

Non-text regions: $Q((1 - \tilde{M}) \otimes \hat{I}_{out}, (1 - \tilde{M}) \otimes I_{bg}),$ (12)

where $\hat{I}_{out, i}$ and $I_{bg, i}$ are the $i$-th text box regions of identically cropped and warped into rectangular images from $\hat{I}_{out}$ and $I_{bg}$, and $n$ is the number of text boxes in the image. The evaluation of text regions indicates how accurately a model localizes text regions and the plausibility of the generated background. The evaluation of non-text regions indicates how robust a model is with regard to text-like textures and how accurately it reconstructs the input. We only evaluated the image quality of two additional regions in the SynthText++ dataset, because the EnsNetSet dataset does not provide text box annotations.

Table 3 shows the performance of our variants when evaluated on the SynthText++ dataset. We evaluated image quality metrics in three different regions: full-image, text, and non-text regions. We investigated the effectiveness of pretraining with text stroke segmentation task, which we referred to as ‘Pretrain’, as mentioned in Section III-B1, and the two-task blending structure as ‘Ours’. We present ‘Single-net’, ‘Ours w/o Pretrain’, ‘Ours’, and for reference, ‘EnsNet’ [1]. We found that ‘Ours’ outperformed ‘Ours w/o Pretrain’, which implies that the prior knowledge from text localization provides useful clues in scene text removal. While the image quality improves gradually in the text-region from ‘Single-net’ to ‘Ours’, there is a greater improvement in image quality between ‘Single-net’ and ‘Ours w/o Pretrain’ in the full-image and non-text regions. This shows...
that the single-net models are poor at reconstruction in non-text regions, whereas ‘Ours’ is robust with regard to text-like textures, as it bypasses non-text regions from the input with explicit wild text localization and selective background inpainting.

Figure 9 shows the qualitative results of EnsNet and our method, with an image difference map to visualize where input image is modified. While EnsNet fails to detect complex text and erases false-positive text-like textures, our method captures and removes text only with plausible backgrounds.

V. CONCLUSION
To address wild-text-related problems in scene text removal, we have designed a novel loss function with a two-task blending structure. The elaborately designed loss function enables the segmentation task to include ambiguous text boundaries appearing frequently in the wild, and the inpainting task selectively restores plausible background only in wild text regions. The proposed method has additional merit, as a wide variety of text location data can be easily utilized in training for robust wild text localization. As validated in the experiments, the proposed method shows outstanding capability in removing wild text regions and outperforms the existing methods. In the future, the proposed method could be efficiently utilized for various applications in the wild.

REFERENCES
[1] S. Zhang, Y. Liu, L. Jin, Y. Huang, and S. Lai, “EnsNet: Ensconce text in the wild,” in Proc. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 801–808.
[2] A. Criminisi, P. Pérez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” IEEE Trans. Image Process., vol. 13, no. 9, pp. 1290–1212, Sep. 2004.
[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “Patch-Match: A randomized correspondence algorithm for structural image editing,” in Proc. ACM SIGGRAPH Papers (SIGGRAPH), Aug. 2009, pp. 1–11.
[4] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan, “Shift-Net: Image inpainting via deep feature rearrangement,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 1–17.
[5] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, “Free-form image inpainting with gated convolution,” 2018. arXiv:1806.03589. [Online]. Available: http://arxiv.org/abs/1806.03589
[6] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 85–100.
[7] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, “ICDAR 2013 robust reading competition,” in Proc. 12th Int. Conf. Document Anal. Recognit., Aug. 2013, pp. 1484–1493.
[8] C. K. Ch’ng, C. S. Chan, and C. Liu, “Total-text: Towards orientation robustness in scene text detection,” Int. J. Document Anal. Recognit., vol. 23, no. 1, pp. 51–52, 2019.
[9] U. Modha and P. Dave, “Image inpainting-automatic detection and removal of text from images,” Int. J. Eng. Res. Appl., vol. 2, no. 2, pp. 930–932, Mar./Apr. 2012. [Online]. Available: https://www.ijera.com
[10] M. Khodadadi and A. Behrad, “Text localization, extraction and inpainting in color images,” in Proc. 20th Iranian Conf. Electr. Eng. (ICEE), May 2012, pp. 1035–1040.
[11] P. D. Wagh and D. R. Patil, “Text detection and removal from image using inpainting with smoothing,” in Proc. Int. Conf. Pervas. Comput. (ICPC), Jan. 2015, pp. 1–4.
[12] M. N. Favorskaya, A. G. Zotin, and M. V. Damov, “Intelligent inpainting system for texture reconstruction in videos with text removal,” in Proc. Int. Congr. Ultra Modern Telecommun. Control Syst., Oct. 2010, pp. 867–874.
[13] Y.-L. Chen, “Automatic text extraction, removal and inpainting of complex document images,” Int. J. Innov. Comput., Inf. Control, vol. 8, pp. 303–327, Jan. 2012.
[14] C. Jung, Q. Liu, and J. Kim, “A new approach for text segmentation using a stroke filter,” Signal Process., vol. 88, no. 7, pp. 1907–1916, Jul. 2008.
[15] A. K. Jain and B. Yu, “Automatic text location in images and video frames,” in Proc. 14th Int. Conf. Pattern Recognit., vol. 2, Aug. 1998, pp. 1497–1499.
[16] M. R. Lyu, J. Song, and M. Cai, “A comprehensive method for multilingual video text detection, localization, and extraction,” IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 2, pp. 243–255, Feb. 2005.
[17] A. Telea, “An image inpainting technique based on the fast marching method,” J. Graph. Tools, vol. 9, no. 1, pp. 23–34, 2004.
[18] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” IEEE Trans. Image Process., vol. 12, no. 8, pp. 882–889, Aug. 2003.
[19] J. Shen and T. F. Chan, “Mathematical models for local nontexture inpainting,” SIAM J. Appl. Math., vol. 62, no. 3, pp. 1019–1043, 2002.
[20] T. Nakamura, A. Zhu, K. Yanai, and S. Uchida, “Scene text eraser,” in Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR), Nov. 2017, pp. 832–837.
[21] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, “EAST: An efficient and accurate scene text detector,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2642–2651.
[22] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, “Character region awareness for text detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9365–9374.
[23] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rivaud, J. Chazallon, W. Khilf, M. M. Luqman, J.-C. Burie, C.-L. Liu, and J.-M. Ogier, “ICDAR2017 robust reading challenge on multi-lingual scene text detection and script Identification–RRC-MLT,” in Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR), Nov. 2017, pp. 1454–1459, doi: 10.1109/ICDAR.2017.237.
[24] 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.
[25] 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.
[26] C. Liu, Y. Liu, L. Jin, S. Zhang, C. Luo, and Y. Wang, “EraseNet: End-to-end text removal in the wild,” IEEE Trans. Image Process., vol. 29, pp. 8760–8775, 2020.
[27] J. Žídek and H. Nakayama, “Erasing scene text with weak supervision,” in Proc. IEEE Winter Conf. Comput. Vis. Appl. Comput. Vis. (WACV), Mar. 2020, pp. 2227–2235.
[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst. (Neurips), vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014, pp. 2672–2680.

| Method       | Full-image | Text region | Non-Text region |
|--------------|------------|-------------|-----------------|
|              | MSE (10^-5) | SSIM        | MSE (10^-5)    | SSIM        | MSE (10^-5) | SSIM        |
| Input Images | 5.55       | 93.84       | 31.80          | 41.91       | 0           | 100         |
| EnsNet [1]   | 0.581      | 95.72       | 2.53           | 70.96       | 0.4192      | 97.12       |
| Single-net   | 0.679      | 94.49       | 2.76           | 64.82       | 0.4971      | 96.38       |
| Ours w/o Pretrain | 0.216  | 98.45       | 2.50           | 72.09       | 0.0564      | 99.78       |
| Ours         | 0.172      | 98.59       | 1.66           | 74.71       | 0.0512      | 99.79       |

TABLE 3. Region-wise image quality analysis. We measured the MSE and SSIM scores from the SynthText++ dataset in three regions: the full-image, text, non-text regions. ‘Pretrain’ means that a model is pretrained with stroke segmentation data as a prior knowledge.
[29] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” 2015, arXiv:1505.04597. [Online]. Available: https://arxiv.org/abs/1505.04597

[30] H. Le and D. Samaras, “Shadow removal via shadow image decomposition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 8578–8587.

[31] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, “GANimation: One-shot anatomically consistent facial animation,” Int. J. Comput. Vis., vol. 128, no. 3, pp. 698–713, Mar. 2020.

[32] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in Proc. IEEE Conf. Comput.Vis. Pattern Recognit., Jun. 2016, pp. 2414–2423.

[33] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 694–711.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[35] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Represent, 2015, pp. 1–14.

[36] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” 2016, arXiv:1611.07004. [Online]. Available: https://arxiv.org/abs/1611.07004

[37] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (ELUs),” 2015, arXiv:1511.07289. [Online]. Available: https://arxiv.org/abs/1511.07289

[38] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. Int. Conf. Mach. Learn., vol. 37, Jul. 2015, pp. 448–456. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045167

[39] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” 2016, arXiv:1607.08022. [Online]. Available: http://arxiv.org/abs/1607.08022

[40] V. Dumoulin, J. Shlens, and M. Kudlur, “A learned representation for artistic style,” 2016, arXiv:1610.07629. [Online]. Available: http://arxiv.org/abs/1610.07629

[41] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8789–8797.

[42] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, “What is wrong with scene text recognition model comparisons? Dataset and model analysis,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 4715–4723.

[43] 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.

JIN YOUNG CHOI (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 1982, 1984, and 1993, respectively. From 1984 to 1989, he was with the Project of TDX Switching System, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. From 1992 to 1994, he was with the Basic Research Department, ETRI, where he was a Senior Member of the Technical Staff involved in the neural information processing systems. Since 1994, he has been with Seoul National University, where he is currently a Professor with the School of Electrical Engineering. From 1998 to 1999, he was a Visiting Professor with the University of California at Riverside, Riverside, CA, USA. He is also with the Automation and Systems Research Institute, the Engineering Research Center for Advanced Control and Instrumentation, and the Automatic Control Research Center, Seoul National University. His current research interests include adaptive and learning systems, visual surveillance, motion pattern analysis, object detection and tracking, and pattern learning and recognition.

DONGYOUN HAN received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2011, 2013, and 2018, respectively. He is currently a Research Scientist with NAVER AI LAB. His current research interests include machine learning and computer vision, in particular deep-learning-based image classification and object detection.

SANGDOO YUN received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2010, 2013, and 2017, respectively. He is currently a Research Scientist with NAVER AI LAB. His current research interests include computer vision, deep learning, and image classification.

BYEONGHO HEO received the bachelor’s and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2012 and 2019, respectively. In 2019, he joined the Clova AI Research, NAVER Corporation, as a Research Scientist, where he is currently working. His current research interests include knowledge distillation, deep learning, image classification, and optimization algorithms.