# Multi-Branch Network with Ensemble Learning for Text Removal in the Wild

Yujie Hou[1], Jiwei Chen[1], and Zengfu Wang[1,2]✉

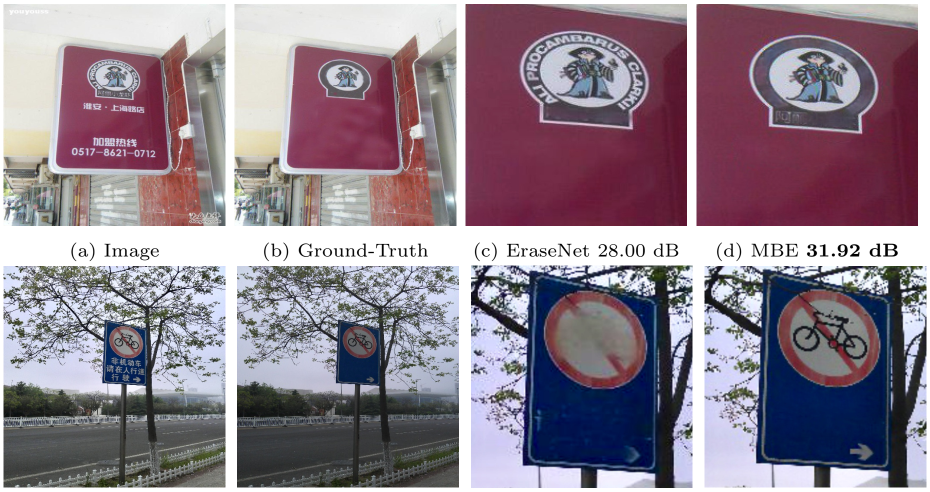
[1] School of Information Science and Technology, University of Science and Technology of China, Hefei, China
[2] Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, China
{houyj1219,cjwbdw06}@mail.ustc.edu.cn, zfwang@ustc.edu.cn

**Abstract.** The scene text removal (STR) is a task to substitute text regions with visually realistic backgrounds. Due to the diversity of scene text and the intricacy of background, earlier STR approaches may not successfully remove scene text. We discovered that different networks produce different text removal results. Thus, we present a novel STR approach with a multi-branch network to entirely erase the text while maintaining the integrity of the backgrounds. The main branch preserves high-resolution texture information, while two sub-branches learn multi-scale semantic features. The complementary erasure networks are integrated with two ensemble learning fusion mechanisms: a feature-level fusion and an image-level fusion. Additionally, we propose a patch attention module to perceive text location and generate text attention features. Our method outperforms state-of-the-art approaches on both real-world and synthetic datasets, improving PSNR by 1.78 dB in the SCUT-EnsText dataset and 4.45 dB in the SCUT-Syn dataset.

## 1 Introduction

The text information that appears in natural scene images is referred to as scene text [16]. Scene text such as license plate numbers and phone numbers may be captured inadvertently when taking a picture [10]. With the development of text detection and recognition technology, such sensitive information could be easily gathered when posting scene images on the Internet. Concealing sensitive information is in high demand to reduce the danger of privacy disclosure. Recent deep learning-based text detection and recognition approaches require a huge number of training data, but manual labeling is time-consuming as well as costly. To fill this gap, the STR is used to generate a synthetic scene text dataset by erasing the text in the image and swapping it for the new text with new content. It might be viewed as a novel method to produce high-quality synthetic datasets for scene text detection and recognition.

Numerous STR techniques in deep learning have been proposed and demonstrated promising performance in recent years (see Sec. 2), yet there still exist multi-scale text inexhaustive erasure problem and the background texture excessive erasure problem in real-world text removal. For example, tiny scene texts



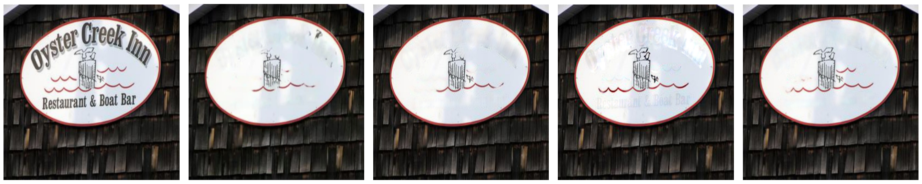
(a) Image (b) Ground-Truth (c) EraseNet 28.00 dB (d) MBE **31.92 dB**

(e) Image (f) Ground-Truth (g) EraseNet 28.02 dB (h) MBE **37.10 dB**

Fig. 1: Examples of text removal in the wild. Qualitative comparisons and quantitive results of PSNR (higher is better) given by the Erasenet [18] and the MBE are shown in the last two columns. Best view with zoom-in.

are difficult to be removed, and symbolic patterns may be deleted incorrectly (see Fig. 1c and Fig. 1g). We analyze previous methods to identify the bottlenecks that restrain the performance. To begin with, a large number of STR methods [2, 16, 28, 37] employ a two-stage paradigm that divides the procedure into a text detection phase and a text removal phase. The divide-and-conquer idea is straightforward, but the two-stage model is vulnerable in a long-range inference. Any mistake that appeared in the text detection process would directly influence the text removal results in the second stage. Second, one STR network is insufficient to capture all text variance in natural scenarios, since different scenes have texts with different fonts, sizes, colors, and illuminations.

To ameliorate the above-mentioned issues, we present a novel framework called Multi-Branch Network with Ensemble Learning (MBE). Rather than dividing the STR into two cascaded steps (text detection stage and text removal stage), we build an ensemble learning model that consists of three parallel STR branches to generate complementary erasure results. The main branch contains a feature fusion high-resolution network (FFHRNet) for retaining complex background texture and the two sub-branches work on hierarchical patch pictures for learning multi-scale semantic features using a U-Net framework. The visualization output of the main branch in Fig. 2d demonstrates that the FFHRNet is capable of keeping high-resolution backgrounds but leaves rough text sketches. The two sub-branches are prone to entirely erasing the text region, but the non-text patterns are also erased (in Fig. 2b and Fig. 2c).



(a) Input (b) Sub-branch 1 (c) Sub-branch 2 (d) Main branch (e) Fused output

Fig. 2: Visualization of intermediate and final results from the MBE. The sub-branch network which extracts multi-scale semantic features is prone to excessively erase non-text regions. The main branch leaves the remains of rough text sketches. The fused result achieves the best performance in erasing scene texts and maintaining the integrity of non-text areas.

Intuitively, combining the complementary outputs from all branches can increase the MBE's stability and performance, resulting in better text-removed results. To do this, we propose two fusion procedures that utilize the strengths of all branches. First, we propose an implicit fusion technique called crossing branch feature fusion (CBF) for fusing multiple branches at the feature level. In terms of CBF, one branch combines several semantic features from other branches. The multi-scale semantic features can enhance the text erasure performance at various font sizes. The patch attention module (PAM) can perceive text locations through a simple segmentation and generate text attention features. Second, a convolutional LSTM module (CLSTM) is proposed as an explicit fusion method to fuse the text-erased results at the image level. The CLSTM can preserve the correct text erasure regions in all outputs while discarding the broken background (refer to Fig. 2e).

Extensive experiments are conducted on both the real-world dataset, SCUT-EnsText [18] and the synthetic dataset, SCUT-Syn [39]. Our MBE outperforms the state-of-the-art method in PSNR by 1.78 dB and 4.45 dB on the two datasets, respectively. The contributions are summarized as follows:

- We propose a multi-branch network with ensemble learning (MBE) for STR, that employs ensemble learning to train three STR branches to learn complimentary erasure outcomes and merges all STR branches via an elaborate fusion process to improve the overall model's reliability and performance.
- The three branches are designed to produce complementary outputs for the model ensemble. The main branch preserves high-resolution information, and two sub-branches learn multi-scale semantic features on patch-level images.
- We propose two fusion strategies to fully exploit the inherent advantages of three branches. An implicit fusion approach assists one branch in fusing semantic and attention features from another branch. An explicit fusion approach combines all branches' suboptimal outputs into the final results.

- The quantitative and qualitative results on both the SCUT-EnsText [18] and the SCUT-Syn [39] datasets indicate that our method outperforms the previous state-of-the-art STR method by a large margin.

## 2 Related Work

Traditional text concealment approaches use image processing to make the text region hard to be detected and recognized. Frome *i.e.* [4] employ a Gaussian filter to blur text region in Google Street View. Inai *i.e.* [10] submerge the scene text by degrading the readability of characters. They use exemplar-based image inpainting to damage the stroke structure. These traditional methods have limitations on complex scenarios, *e.g.* complicated backgrounds, or perspective distortion, and they fail to fill the text region with a plausible background.

Inspired by the notable success of deep learning, novel methods are proposed to conceal the scene text. Nakamura *i.e.* [24] first design an automatic scene text eraser that converts the text erasure problem into an image transformation problem. They use neural networks to learn the transformation function between the scene text images and non-text images without the annotations for the text locations. EnsNet [39] uses a lateral connection structure to integrate high-level semantics and low-level spatial information. Four loss functions are proposed to improve the reality of filling backgrounds. The early methods are lightweight as well as fast but often left noticeable text remnants.

A couple of two-step approaches decouple the scene text removal into two sub-tasks: the text detection task and the text removal task [2,16,28,37]. The text position predicted in the first stage guides the second stage where to erase, and the then text region is filled with a meaningful background. However, if text detection is incorrect, it will degrade the text removal results by leaving text remnants or breaking the integrity of the non-text region. Thus, two-stage STR methods might cause apparent drawbacks and generate low-quality text-removed images. Benefiting from the development of generative adversarial network (GAN) [6,15,22], image restoration methods achieve significant improvement in generating local textures. Erasenet [18] proposes a coarse-to-fine generator network with a local-global discriminator network. In [2], they connect two GAN-based frameworks to refine text removal results at the stroke level. Though the GAN-based methods might improve the erasure quality, the training process is not stable and heavy.

## 3 Multi-Branch Network with Ensemble Learning

### 3.1 Overall Architecture

The motivation of our method is that different STR branches can produce complementary text removal results, and it is reasonable to combine them for a better text-removed image. To acquire complementary results, we propose three text removal branches in our method. The main branch (FFHRNet) preserves fine

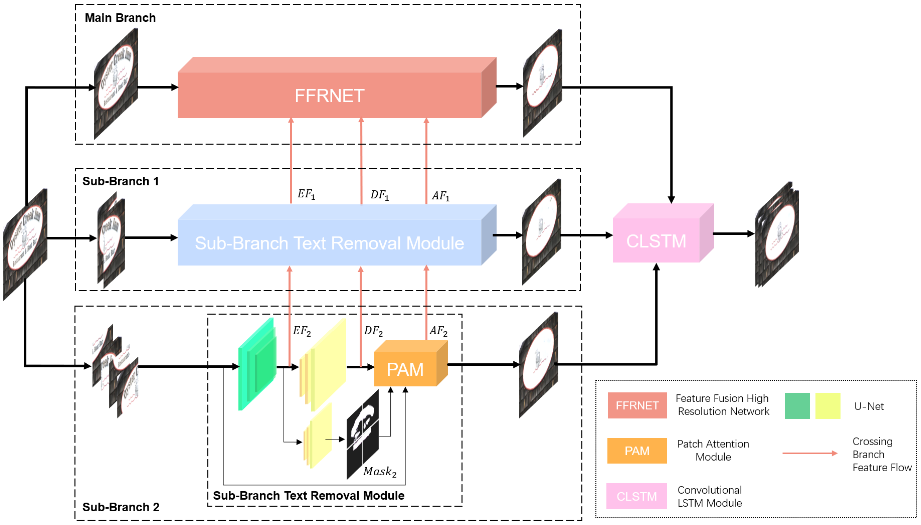

Main Branch
FFRNET
Sub-Branch 1
$EF_1$
$DF_1$
$AF_1$
Sub-Branch Text Removal Module
CLSTM
$EF_2$
$DF_2$
$AF_2$
PAM
$Mask_2$
Sub-Branch Text Removal Module
Sub-Branch 2
FFRNET Feature Fusion High Resolution Network
U-Net
PAM Patch Attention Module
Crossing Branch Feature Flow
CLSTM Convolutional LSTM Module

Fig. 3: An overview of our model. The black bold arrows represent the main branch, sub-branch 1, and sub-branch 2, respectively. Sub-branch 1 uses two non-overlapping patch images as input and sub-branch 2 uses four non-overlapping patch images as input. The red arrows represent the crossing branch feature fusion. $AF$: Attention Feature. $EF$: Encoder Feature. $DF$: Decoder Feature.

textures from the original resolution image. The sub-branches operate on patch-hierarchical images for learning multi-scale semantic features via an encoder-decoder architecture. Following that, we combine the three branches with the CFB as a feature fusion module and the CLSTM as an image fusion module for the model ensemble.

### 3.2 Main Branch Network

Exiting STR approaches [2, 18, 32, 39] almost follow the U-Net framework. Due to the repetitive usage of scale variation procedures in U-Net, the text-erased results are prone to damage the integrity of the background and details of local texture from the original picture. Thus, our main branch employs the feature fusion high-resolution network (FFHRNet), which is inspired by HRNet [5, 27] and the channel attention mechanism [40], to retain high-resolution representations. A high-resolution branch (the black bold arrow in the Fig. 4) and three downsampling branches make up the FFHRNet. The high-resolution branch includes no downsampling or upsampling operations to generate texturally-enriched representations and gradually adds low-resolution feature representations from lower branches to fuse multi-scale information. To adaptively re-calibrate the feature map in the channel dimension, channel attention blocks [40] have been implemented in the high-resolution branch.

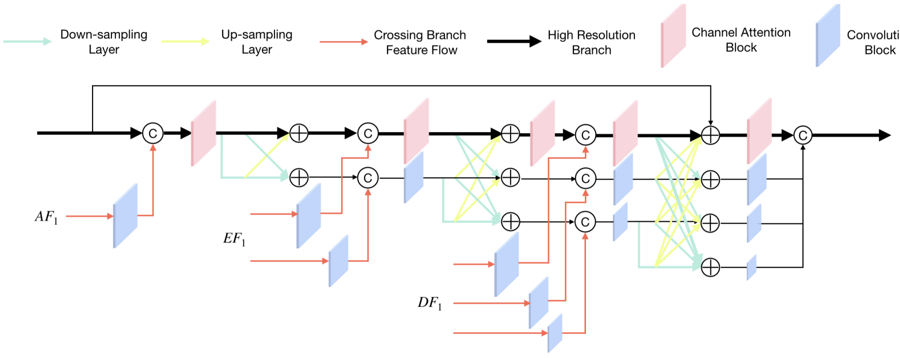


Fig. 4: The feature fusion high-resolution network (FFHRNet). The red-bold arrows illustrate how and where to fuse features from sub-branch 1.

Attention features ($AF_1$), encoder features ($EF_1$) and decoder features ($DF_1$) from sub-branch 1 are gradually transferred to the FFHRNet, according to the crossing branch feature fusion process (see Sec. 3.4). The $AF_1$, which carries text location information, is concatenated with feature maps in the high-resolution branch after being scaled by a convolution block. The $EF_1$ and $DF_1$ are fused with FFHRNet in the same way.

### 3.3 Sub-Branch Network

**Encoder-Decoder network.** The two sub-branch networks, which mainly contain an encoder-decoder network [25] and a PAM, capture the text in scene images and wipe them as precisely as possible. Scene text in the wild has varied scales, which may give rise to failures in text detection, so we separate the input image into several non-overlapping patches to develop a multi-patch hierarchy learning model, which is akin to [26, 36, 38]. The patch-level inputs compared to the original image considerably lighten the sub-branch network and naturally provide multi-scale semantic information. We also utilize the skip connection technique [19] in the encoder-decoder network to link the downsampling layer and the upsampling layer to reduce information loss. In the U-Net module, the patch images first map to low-resolution representations and then progressively recover to the original resolution by applying several reverse mappings. Annotations for text location are not available in the model inference phase, thus we train another decoder module [18] after the encoder module of U-Net to determine text positions. The binary masks ($Mask_1$ and $Mask_2$) roughly segment scene texts and non-text background (0 for text and 1 for background), which is supervised by dice loss Eq. (3). The PAM then uses the text mask to compute the attention feature.

**Patch Attention Module (PAM).** The goal of PAM is to enhance the text position response in the feature map. As illustrated in Fig. 5, we element-wise

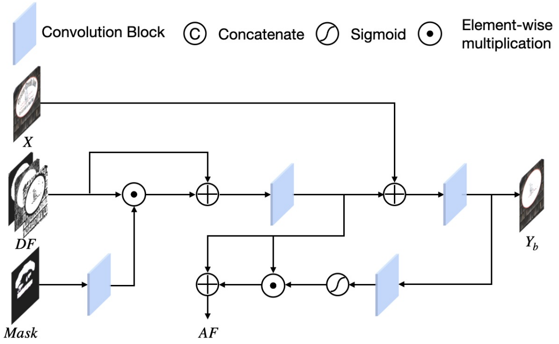


Fig. 5: The architecture of patch attention module (PAM).

multiply the input decoder feature $DF \in \mathbb{R}^{H\times W\times C}$ from early encoder-decoder network with the binary mask $M \in \mathbb{R}^{H\times W\times 1}$ where $H$, $W$ denotes the height and width dimension of feature map, and $C$ is the number of channel. Then, the new feature map containing both multi-scale semantic features and text location information is added to the original input $X$ as a residual part to obtain the text removal output $Y_b$ in the sub-branch. To get the attention feature $AF$, we import the output $Y_b$ to a convolution layer followed by the sigmoid activation, and then we element-wise multiply the activated feature map with the previous feature map as an attention-guided residual, which is added back to the previous feature map. The attention mechanisms could suppress the less informative features and only retain the useful parts. Finally, the attention feature representation $AF$ will be passed to other branches for information fusion.

### 3.4 Crossing Branch Feature Fusion (CBF)

The CBF is presented as a method for implicitly fusing three STR branches at the feature level. From the main branch (top) to the sub-branch (bottom), the input image is gradually split into smaller patches, and we fuse the features in a down-top pathway to merge low-resolution, semantically strong features with high-resolution, semantically weak features. The red arrows in Fig. 3 indicate two paths of feature transmission: from sub-branch 2 to sub-branch 1 ( $\text{CBF}_1$), and from sub-branch 1 to the main branch ($\text{CBF}_2$). The attention feature $AF_2$ from sub-branch 2 concatenates with the patch-level input image $X_1$ after convolution block resizing, and the encoder and decoder feature $EF_2, DF_2$ are imported as supplementary inputs into the U-Net in sub-branch 1. We transmit the feature maps from all encoder layers and decoder layers in the U-Net of sub-branch 2, which means $EF_2 = \{EF_{2a}, EF_{2b}, EF_{2c}\}$, $DF_2 = \{DF_{2a}, DF_{2b}, DF_{2c}\}$. In $\text{CBF}_1$, the multi-scale semantic features from sub-branch 2 are progressively imported to the same-scale encoder-decoder layers in sub-branch 1 to improve the scene text erasure performance at any text size. In $\text{CBF}_2$, the encoder and decoder features $EF_1$, $DF_1$ and the attention feature $AF_1$ from sub-branch 1 are transmitted to the main branch following the same setting in $\text{CBF}_1$.

The method of feature fusion across all STR branches has several advantages. First, it reduces the vulnerability of the feature by repeated use of up- and down-sampling operations in the encoder-decoder network, and the results are more robust when the model confronts information loss due to various interferences. Second, the feature of one branch containing multi-scale semantic information and text location information can help enrich the feature of the other branches. Third, the network optimization procedure becomes more stable as it eases the flow of information, thereby allowing us to train three complementary STR networks and the CLSTM successfully without a model collapse. Additionally, the ablation experiment in Sec. 4.3 demonstrates the effectiveness of CBF in text erasing performance.

### 3.5 Convolutional LSTM module (CLSTM)

The CLSTM module [9, 35] is another fusion mechanism to explicitly merge multiple branches at the image level. We stack the text removal results from three branches into a sequence pattern, and the CLSTM could predict the next images of the sequence data as our fused output. Due to the gate cell mechanism, the fused outputs can retain correct erasure results from complementary branches while discarding the incorrect erasure sections.

### 3.6 Train Strategy

For training our MBE, we use the scene text image $X$, text-removed ground-truth $I_{gt}$, and text mask ground-truth $M$ as inputs. We optimize our method with the Charbonnier loss [3, 21], Edge loss, and Dice loss. The details of our loss functions are as follows.

Because of the multi-branch framework, the Charbonnier loss and Edge loss penalize all branch outputs. The subscript $i$ (from 1 to 3) in Eq. (1) and Eq. (2) denotes the main branch, sub-branch 1, and sub-branch 2, respectively. The Charbonnier loss is defined as:

$$L_{Char} = \sum_{i=1}^{3} \sqrt{\left\| Y_i - I_{gt} \right\|^2 + \varepsilon^2} \tag{1}$$

where $Y_i$ represents the text-erased outputs from different branch networks. The Charbonnier loss has a constant parameter $\varepsilon$, which is set to $10^{-3}$ in our experiments.

The Edge loss is defined as:

$$L_{Edge} = \sum_{i=1}^{3} \sqrt{\left\| \triangle Y_i - \triangle I_{gt} \right\|^2 + \varepsilon^2} \tag{2}$$

where $\triangle$ denotes the Laplacian operator. $\varepsilon$ is set to $10^{-3}$ in our experiments.

Table 1: Ablation study on different components of the proposed MBE on SCUT-EnsText. FFHRNet: feature fusion high-resolution network. SB$_1$: sub-branch network 1. SB$_2$: sub-branch network 2. CBF$_1$: crossing branch feature fusion (SB$_2$ to SB$_1$). CBF$_2$: crossing branch feature fusion (SB$_1$ to FFHRNet). PAM: patch attention module. CLSTM: convolutional LSTM module.

| FFHRNet | SB$_1$ | SB$_2$ | CBF$_1$ | CBF$_2$ | PAM | CLSTM | PSNR(↑) | MSSIM(↑) | AGE(↓) | pEPs(↓) | pCEPs(↓) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | 31.2012 | 0.9575 | 2.8951 | 0.0235 | 0.0162 |
| ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | 34.2082 | 0.9690 | 2.1542 | 0.0146 | 0.0098 |
| ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | 34.3908 | 0.9699 | 2.1129 | 0.0140 | 0.0094 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | 34.5285 | 0.9684 | 2.1059 | 0.0142 | 0.0094 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | 34.6652 | 0.9705 | 2.0835 | 0.0139 | 0.0094 |
| ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | 34.7209 | 0.9708 | 2.0877 | 0.0134 | 0.0090 |
| ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | 34.7084 | 0.9710 | 2.1162 | 0.0136 | 0.0093 |
| ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | 34.2236 | 0.9698 | 2.1854 | 0.0147 | 0.0107 |
| ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | 34.3568 | 0.9705 | 2.1235 | 0.0135 | 0 0092 |
| ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | 34.7896 | 0.9730 | 2.0594 | **0.01239** | **0.0083** |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | 34.2552 | 0.9686 | 2.2819 | 0.0153 | 0.0106 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | 34.6652 | 0.9705 | 2.0835 | 0.0139 | 0.0094 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | **35.0304** | **0.9731** | **2.0232** | 0.01282 | 0.0088 |

The Dice loss in Eq. (3) is proposed for mask segmentation to learn the text location. It measures the proportion of correctly predicted pixels to the sum of the total pixels of both prediction and ground truth. The Dice loss can be formulated as:

$$L_{Dice} = 1 - \frac{2\sum_{x,y}(S_{x,y}) \times (M_{x,y})}{\sum_{x,y}(S_{x,y})^2 + \sum_{x,y}(M_{x,y})^2} \tag{3}$$

where $S$ represents the predicted mask from the decoder module and $M$ represents the text mask ground-truth. $S_{x,y}$ and $M_{x,y}$ denote the pixel value at point $x, y$.

Finally, we sum three loss functions together to form the total loss of our MBE, which is defined as Eq. (4):

$$L_{Total} = L_{Char} + \lambda L_{Edge} + \mu L_{Dice} \tag{4}$$

where the $\lambda$ is set to 0.05 and $\mu$ is set to 0.105.

## 4 Experiments and Analysis

### 4.1 Datasets and Evaluation Protocol

**Datasets.** We use the images from SCUT-EnsText [18] and SCUT-Syn [39] dataset to train our method. These two datasets are widely applied in previous STR methods, and we test our approach by following the same method in [18]. The SCUT-EnsText is a real-world dataset including 2,749 training images and

Table 2: Quantitative comparison of our method and start-of-the-art methods on SCUT-EnsText and SCUT-Syn datasets. The Best and second best scores are highlighted and underlined, respectively.

| Methods | SCUT-EnsText | | | | | SCUT-Syn | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR(↑) | MSSIM(↑) | AGE(↓) | pEPs(↓) | pCEPs(↓) | PSNR(↑) | MSSIM(↑) | AGE(↓) | pEPs(↓) | pCEPs(↓) |
| Bain [2] | 15.9399 | 0.5706 | 24.8974 | 0.3282 | 0.2123 | 20.83 | 0.8319 | 10.5040 | 0.1021 | 0.5996 |
| Pixel2Pixel [11] | 26.6993 | 0.8856 | 6.0860 | 0.0480 | 0.0337 | 26.67 | 0.9108 | 5.4678 | 0.0473 | 0.0244 |
| SceneTextEraser [24] | 25.4651 | 0.9014 | 6.0069 | 0.0532 | 0.0296 | 25.40 | 0.9012 | 9.4853 | 0.0553 | 0.0347 |
| EnsNet [39] | 29.5382 | 0.9274 | 4.1600 | 0.0307 | 0.0136 | 37.36 | 0.9644 | 1.7300 | 0.0069 | 0.0020 |
| Zdenek [37] | - | - | - | - | - | 37.46 | 0.9810 | - | - | - |
| MTRNet [29] | - | - | - | - | - | 29.71 | 0.9443 | - | - | - |
| MTRNet++ [28] | 24.6145 | 0.8990 | 11.3669 | 0.1459 | 0.0869 | 34.55 | 0.9845 | - | - | - |
| EraseNet [18] | 32.2976 | 0.9542 | 3.1264 | 0.0192 | 0.0110 | 38.32 | 0.9765 | 1.5982 | 0.0048 | 0.0004 |
| PERT [32] | 33.2492 | 0.9695 | 2.1833 | 0.0136 | **0.0088** | 39.40 | 0.9787 | 1.4149 | 0.0045 | 0.0006 |
| **MBE (Ours)** | **35.0304** | **0.9731** | **2.0594** | **0.01282** | **0.0088** | **43.85** | **0.9864** | **0.9356** | **0.0013** | **0.00004** |

813 test images, and the SCUT-Syn is a synthetic dataset containing 8000 training images and 800 test images with a size of 512×512. A scene text image is processed into a 1-2-4 multi-patch model for multi-branch input shown in Fig. 3. The notation 1-2-4 indicates the number of non-overlapping image patches from the coarsest level to the finest level.

**Evaluation Protocol.** We adopt the same evaluation metrics in [18, 39] to comprehensively evaluate the text erasure performance, which includes peak signal to noise ratio (PSNR), multi-scale structural similarity (MSSIM) [33], an average of the gray level absolute difference (AGE), percentage of error pixels (pEPs) and percentage of clustered error pixels (pCEPS).

### 4.2 Implementation Details

The MBE is an end-to-end trained model implemented in Pytorch. Vertical and horizontal flips are randomly applied as data augmentation. We use the Adam optimizer [17] with an initial learning rate of $2 \times 10^{-4}$ and apply the cosine annealing strategy [20] to steadily decrease the learning rate to $1 \times 10^{-6}$.

### 4.3 Ablation Study

In this subsection, we investigate the effect of each component in our proposed MBE step by step. The number of branch networks, the cross branch feature fusion (CBF), the patch attention module (PAM), and the convolutional LSTM module (CLSTM) are the focus of our study. Evaluations are performed on the real-world SCUT-EnsText dataset. The quantitative results are shown in Tab. 1 which demonstrates that the MBE outperforms all inferior models.

**Number of branch networks.** This subsection identifies the number of branch networks that perform optimally. We gradually increase the number of sub-branches from 0 to 3. The sub-branch 3 takes eight non-overlapping patch images as input. By integrating sub-branch 1 and sub-branch 2 into the FFHRNet, our model achieves a higher performance than the previous model demonstrated in the Tab. 1. When the sub-branch 1 or sub-branch 2 modules are

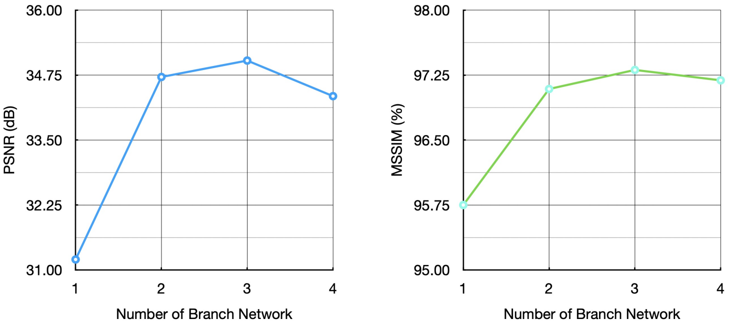


Fig. 6: PSNR *vs.* Number of branch networks and MSSIM *vs.* Number of branch networks. Our method achieves the best performance when we set three branches.

removed from our final model, the inferior model's PSNR performance decreases from 35.03 dB to 34.72 dB and 34.70 dB, respectively. However, when we continue to insert the sub-branch into the overall model, both PSNR and MISSM deteriorate (Fig. 6). This demonstrates that we cannot fuse branch networks infinitely to enhance the model's performance. Excessive branches would obstruct the ensemble model and need extra calculations during training. Because a network with a large number of branches is hard to train as a multi-task learning, the entire model is not stable during optimization. As a result, the generated outcomes are worse than before. In brief, our method employs a single main branch and two sub-branches.

**Cross Branch Feature Fusion.** To demonstrate the effect of our proposed CBF on model fusion, we divide the CBF into two components ($\mathrm{CBF}_1$ and $\mathrm{CBF}_2$) and test them separately. $\mathrm{CBF}_1$ combines the features of sub-branch 1 with features of sub-branch 2, and $\mathrm{CBF}_2$ combines the features of the main branch with the features of sub-branch 1. The ablation research in Tab. 1 shows that the PSNR climbs from 34.20 dB to 34.39 dB and subsequently to 34.52 dB when we gradually add the $\mathrm{CBF}_1$ and $\mathrm{CBF}_2$ module to the model. When either $\mathrm{CBF}_1$ or $\mathrm{CBF}_2$ is removed from our final model, the inferior model performs poorly on most of the criteria. The two ablation studies demonstrate that both the two feature fusion mechanisms can significantly improve STR performance.

**PAM and CLSTM.** We remove the PAM in both sub-branches to verify the effectiveness of text location information for text removal. In the Tab. 1, we observe that the previous model produces better results after adding the PAM and the MBE model decreases by a large margin when the PAM is removed. The CLSTM combines text-erased outputs from all branch networks. Our model with CLSTM has a considerable improvement of 0.36 dB in PSNR compared to the model without it. The visualization results in Fig. 2d show that the fused output contains both the text-erased results in the sub-branch outputs and the integrity of the background in the main branch output.

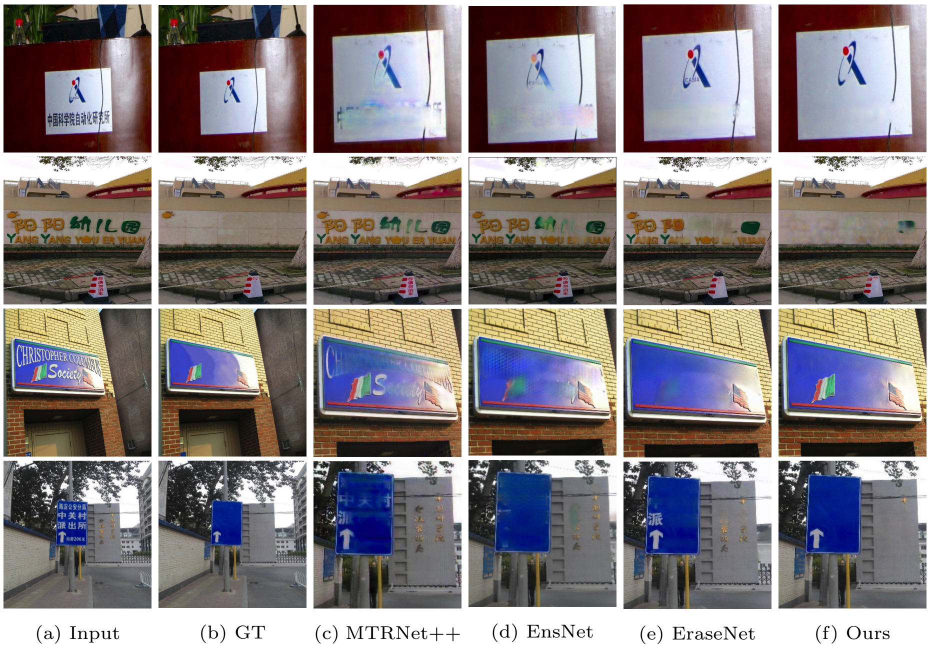

(a) Input (b) GT (c) MTRNet++ (d) EnsNet (e) EraseNet (f) Ours

Fig. 7: Examples of text erased image on SCUT-EnsText of comparing MBE with previous methods. Best viewed with zoom-in.

### 4.4 Comparison with State-of-the-Art Methods

In this section, we verify the effectiveness of MBE by comparing our method against recent state-of-the-art approaches on the SCUT-EnsText and the SCUT-Syn datasets. The results are shown in Tab. 2. The MBE almost achieves the best performance in all metrics on two datasets. Compared to the current best STR method, PERT [32], we obtain a performance gain of 1.78 dB on the SCUT-EnsText dataset and 4.45 dB on the SCUT-Syn dataset in PSNR.

Fig. 7 illustrates visualization results to compare our MBE with other methods. MTRNet++ [28] and EnsNet [39] cannot exhaustively remove text with arbitrary orientations or irregular fonts, and cause the outputs with rough text sketches. EraseNet [18] has higher text removal performance than MTRNet++ and EnsNet, but some non-text patterns are wiped because of failures in text detection. In addition, small-size texts can not be perfectly removed. Our method alleviates the aforementioned problems. Scene texts in various scales are removed, and the complex backgrounds are reserved.

### 4.5 The Effectiveness of Complementary Multi-Branch Network

The three branches in MBE are designed for learning complementary text-removal functions so that the final fused results achieve the best performance

Table 3: Ablation study on different modules in multi-branch network.

| Multiple Branch Framework | PSNR(↑) | MSSIM(↑) |
|---|---|---|
| ResNet-101 + UNet + UNet | 34.2614 | 0.9657 |
| FFHRNet + ResNet101 + UNet | 34.4316 | 0.9711 |
| FFHRNet + UNet + ResNet-101 | 33.5561 | 0.9678 |
| ResNet-101+ResNet-101+ResNet-101 | 32.1547 | 0.9452 |
| **FFHRNet + UNet + UNet** | **35.0304** | **0.9731** |

Table 4: Robustness study on SCUT-EnsText with various degraded inputs.

| Degradation | EraseNet [18] | | MBE | |
|---|---|---|---|---|
| | PSNR(↑) | MSSIM(↑) | PSNR(↑) | MSSIM(↑) |
| Blur | 25.92 | 0.80 | 26.04 | 0.82 |
| Noise | 23.91 | 0.70 | 24.85 | 0.78 |
| Rain | 27.51 | 0.87 | 28.04 | 0.90 |

compared to the output from the individual STR branch. To verify the effectiveness of the complementary network mechanism, the FFHRNet in the main branch and U-Net in the sub-branch are replaced with another strong benchmark, ResNet-101 [8] to form an irrelevant multi-branch network. We compare four variations of the multi-branch framework: ResNet-101 in the main branch, ResNet-101 in sub-branch 1, ResNet-101 in sub-branch 2, and ResNet-101 in all branches.

Experimental results with various multi-branch networks are shown in Tab. 3. We observe that the complementary framework in MBE achieves the best performance compared to all variations of the multi-branch framework, even though the ResNet-101 has a stronger performance than the U-Net. It implies that simply fusing multiple modules instead of considering the relationship among multiple branches can not enhance the text removal results. As a brief conclusion, we use the FFHRNet and U-Net as a backbone to construct the multiple complementary branches framework for ensemble learning.

### 4.6 Robustness Analysis

We demonstrate our method's robustness in this experiment with three degraded image approaches on the SCUT-EnsText: blurry image, noisy image, and rainy image. To create blurry photos, we use a Gaussian filter with a kernel size of 5×5. We add Gaussian noise with a mean of 0.5 and a variation of 0.1 to generate noisy photos. To create rainy images, we multiplied the length of Gaussian noise by 10 and rotated them 45 degrees to simulate the direction of rain in nature.

As shown in Tab. 4, it demonstrates our method achieves higher performance in both PSNR and MSSIM than EraseNet. This proves that the ensemble learning in our method can improve the model's robustness when facing interference factors and information loss.

### 4.7 Synthetic data via STR for Scene Text Recognition

Since MBE can provide reliable performance on text removal, we extend it to generate a synthetic dataset via replacing the background inpainting module in the SRNet [34] with MBE (SR-MBENet). The SRNet can replace a text in the source image with another one while retaining the styles of both the background and the original text. We collect 50000 real-world data from [13, 14, 23, 30] as

Table 5: Scene text recognition accuracy results on 4 benchmark test datasets. We train DTRB with three synthetic datasets.

| Model | Train Data | IC13 [14] | IC15 [13] | IIIT [23] | SVT [31] |
|---|---|---|---|---|---|
| DTRB [1] | MJSyn [12]+SynText [7] | **93.6** | 77.6 | 87.9 | **87.5** |
| DTRB [1] | Synth-data 1 (SRNet) | 87.2 | 64.0 | 84.6 | 77.1 |
| DTRB [1] | Synth-data 2 (SR-MBENet) | 93.1 | **78.2** | **89.5** | 86.7 |

style images and generate 20 text-swapped images from a single style image to construct our synthetic dataset via SR-MBENet. We train the scene text recognition [1] with three synthetic datasets to analyze the effect of our method by the recognition performance and follow the same evaluation protocol in [1].

In Tab. 5, we find that the synthetic dataset generated by SRNet might decrease the model performance compared to the baseline (MJSyn [12] +SynText [7]). The reason for it might be that the texts with rare font shapes failed to transfer to new text or the complex text structures are hard to erase leading to noise labels. The SR-MBENet can improve the quality of synthetic datasets by alleviating the second problem. Thus, the model trained with our proposed dataset achieves the highest performance on two benchmarks.

## 5 Conclusion

We propose a novel method for the STR task by training multiple complementary STR models and combining them for a better result with ensemble learning to solve the multi-scale text erasure problem and background destruction problem. Combining a diverse set of individual STR models can also improve the stability of the overall model, leading to more reliable results than individual models. To ensure synergy between reciprocal branches, we propose a crossing branch feature fusion guideline to help features flow in all branches. The intermediate outputs from different branches are fused in a fusion module for final results. Our model achieves significant performance gains compared to previous STR methods on both the real-world dataset and the synthetic dataset. In the future, we will extend the MBE to a novel scene text editing method that can swap text in scene images with another one while maintaining a realistic look. We believe the new synthetic dataset can fill the gap of shortage in a reliable, large-scale scene text dataset.

**ACKNOWLEDGMENTS** This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDC08020400).

## References

1. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is wrong with scene text recognition model comparisons? dataset and model analysis.

In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4715–4723 (2019) 14

2. Bian, X., Wang, C., Quan, W., Ye, J., Zhang, X., Yan, D.M.: Scene text removal via cascaded text stroke detection and erasing. arXiv preprint arXiv:2011.09768 (2020) 2, 4, 5, 10
3. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: Proceedings of 1st International Conference on Image Processing. vol. 2, pp. 168–172. IEEE (1994) 8
4. Frome, A., Cheung, G., Abdulkader, A., Zennaro, M., Wu, B., Bissacco, A., Adam, H., Neven, H., Vincent, L.: Large-scale privacy protection in google street view. In: 2009 IEEE 12th international conference on computer vision. pp. 2373–2380. IEEE (2009) 4
5. Geng, Z., Sun, K., Xiao, B., Zhang, Z., Wang, J.: Bottom-up human pose estimation via disentangled keypoint regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14676–14686 (2021) 5
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014) 4
7. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2315–2324 (2016) 14
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 13
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997) 8
10. Inai, K., Pålsson, M., Frinken, V., Feng, Y., Uchida, S.: Selective concealment of characters for privacy protection. In: 2014 22nd International Conference on Pattern Recognition. pp. 333–338. IEEE (2014) 1, 4
11. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017) 10
12. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 (2014) 14
13. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al.: Icdar 2015 competition on robust reading. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 1156–1160. IEEE (2015) 13, 14
14. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 1484–1493. IEEE (2013) 13, 14
15. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017) 4
16. Keserwani, P., Roy, P.P.: Text region conditional generative adversarial network for text concealment in the wild. IEEE Transactions on Circuits and Systems for Video Technology (2021) 1, 2, 4
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 10

18. Liu, C., Liu, Y., Jin, L., Zhang, S., Luo, C., Wang, Y.: Erasenet: End-to-end text removal in the wild. IEEE Transactions on Image Processing **29**, 8760–8775 (2020) 2, 3, 4, 5, 6, 9, 10, 12, 13
19. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015) 6
20. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 10
21. Mehri, A., Ardakani, P.B., Sappa, A.D.: Mprnet: Multi-path residual network for lightweight image super resolution. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2704–2713 (2021) 8
22. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014) 4
23. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: BMVC-British machine vision conference. BMVA (2012) 13, 14
24. Nakamura, T., Zhu, A., Yanai, K., Uchida, S.: Scene text eraser. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 832–837. IEEE (2017) 4, 10
25. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015) 6
26. Suin, M., Purohit, K., Rajagopalan, A.: Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3606–3615 (2020) 6
27. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5693–5703 (2019) 5
28. Tursun, O., Denman, S., Zeng, R., Sivapalan, S., Sridharan, S., Fookes, C.: Mtrnet++: One-stage mask-based scene text eraser. Computer Vision and Image Understanding **201**, 103066 (2020) 2, 4, 10, 12
29. Tursun, O., Zeng, R., Denman, S., Sivapalan, S., Sridharan, S., Fookes, C.: Mtrnet: A generic scene text eraser. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 39–44. IEEE (2019) 10
30. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 (2016) 13
31. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International conference on computer vision. pp. 1457–1464. IEEE (2011) 14
32. Wang, Y., Xie, H., Fang, S., Qu, Y., Zhang, Y.: A simple and strong baseline: Progressively region-based scene text removal networks. arXiv preprint arXiv:2106.13029 (2021) 5, 10, 12
33. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004) 10
34. Wu, L., Zhang, C., Liu, J., Han, J., Liu, J., Ding, E., Bai, X.: Editing text in the wild. In: Proceedings of the 27th ACM international conference on multimedia. pp. 1500–1508 (2019) 13
35. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems. pp. 802–810 (2015) 8

36. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14821–14831 (2021) 6
37. Zdenek, J., Nakayama, H.: Erasing scene text with weak supervision. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2238–2246 (2020) 2, 4, 10
38. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5978–5986 (2019) 6
39. Zhang, S., Liu, Y., Jin, L., Huang, Y., Lai, S.: Ensnet: Ensconce text in the wild. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 801–808 (2019) 3, 4, 5, 9, 10, 12
40. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 286–301 (2018) 5