Erasing Scene Text with Weak Supervision

Jan Zdenek Hideki Nakayama
The University of Tokyo
{jan, nakayama}@nlab.ci.i.u-tokyo.ac.jp

Abstract

Scene text erasing is a task of removing text from natural scene images, which has been gaining attention in recent years. The main motivation is to conceal private information such as license plate numbers, and house nameplates that can appear in images. In this work, we propose a method for scene text erasing that approaches the problem as a general inpainting task. In contrast to previous methods, which require pairs of original images containing text and images from which the text has been removed, our method does not need corresponding image pairs for training. We use a separately trained scene text detector and an inpainting network. The scene text detector predicts segmentation maps of text instances which are then used as masks for the inpainting network. The network for inpainting, trained on a large-scale image dataset, fills in masked out regions in an input image and generates a final image in which the original text is no longer present. The results show that our method is able to successfully remove text and fill in the created holes to produce natural-looking images.

Figure 1: Examples of scene text erasing. Original images are on the left, images where the text was erased by our model on the right.

1. Introduction

Nowadays, text is present almost everywhere around us and is an inseparable part of our daily lives. Text is used as a way to communicate and convey various kinds of information. When taking photos for personal enjoyment or for the purpose of collecting data, some text is inevitably bound to appear in the images, whether captured intentionally or incidentally. However, a lot of text contains personal and private information, e.g. vehicle registration plates, ID numbers, names, and home addresses, which could be misused if it became publicly available on the internet. To prevent that, it is desirable to remove any exploitable information from images before releasing them to the public, which calls for a method to automatically erase text from images without severely degrading their visual quality. The problem can be divided into two subtasks:

1. detecting and removing pixels that belong to text regions

2. filling missing pixels to naturally blend with the background

Previous research on scene text erasing approaches the problem as supervised image-to-image translation. However, that requires to prepare images containing text instances and their corresponding counterparts without text for training, which is difficult and expensive for real-life scene images. Zhang et al. [39] partly avoid this problem by using synthetic data for training, but such data semantically deviates from real-life scenes and creates a bias that might lead to a drop in performance when transferring the model to images of real-life scenes.

This work is the first to propose a method for scene text erasing that does not require training data pairs of original

images with text and their corresponding ground-truth images where the text has been removed. We achieve it by abstracting the problem as a general inpainting task and use a trained scene text detector to predict which parts of the image contain text and need to be removed (subtask 1) and repaint those parts using an inpainting network (subtask 2). Training a scene text detector requires less expensive annotation than training a scene text eraser using the aforementioned corresponding image pairs because only bounding boxes of text regions are required. The inpainting network is trained on general natural scene images with random masks and therefore it is expected to be suitable for filling backgrounds. As we do not use the target images without text for training, we consider our method as requiring weaker supervision compared to previous works. Example of results obtained by our proposed method are shown in Figure 1.

2. Related Work

2.1. Text Erasing

Early research on erasing text from images focused on removing captions and subtitles from video frames [19, 25, 17]. Spatial restoration in the current frame and temporal restoration in consecutive frames was adopted in [19, 25] to replace pixels of text. Khodadadi et al. [17] used a pattern matching algorithm to replace text pixels. Recently, Kim et al. [18] proposed a method for video decaptioning that uses an encoder-decoder neural network with multi-frame input, exploiting spatio-temporal information in the video. However, these methods focus on digitally created text in images and require the text to be well-aligned, clean, and in focus, which is not always the case in scene text images due to their complexity, distortion, different lighting conditions, etc.

Nakamura et al. [26] were the first to propose a method for erasing text from natural scene images. They divide the image into small patches using a sliding windows method and use a U-Net [30] shaped neural network to erase the text. As the erasing process is performed separately on all patches, which are then merged back together, the results tend to be inconsistent.

Recent methods for scene text erasing [39, 33] are based on generative adversarial networks (GANs) [5]. In particular, they are derived from the family of GANs that perform transformations on an image, such as image-to-image translation [12, 43]. Zhang et al. [39] employ a GAN with a U-Net shaped encoder-decoder generator, and train it by using several loss functions, namely content loss [14], texture loss [38], total variation loss [14], and multi-scale regression loss. In addition, they propose a local-aware patch-based discriminator that only penalizes the patches that contain text. Tursun et al. [33] use a ground-truth text region mask as an additional input into the network. This enables selective erasing by allowing the users to select text regions that they want to erase. However, a disadvantage of this method is that a text region mask is always required, which means that text removal cannot be performed fully automatically without user guidance. Without any input mask, the model can still erase some text but the performance is very limited. Note that our proposed method can also be employed for selective erasing if we allow users to select which predetected text regions they want to erase.

All of the methods for scene text erasing mentioned above require pairs of original images containing text and ground-truth images with the text removed for training. Obtaining such data is a difficult and expensive process. One way of achieving that, used in [39], is to utilize photo-editing software and manually edit the original images. However, removing text from complex scene images so that the result looks naturally requires a skilled person and a lot of time. Another way to obtain ground-truth images without text is to use conventional inpainting methods, as performed in [26]. To alleviate the problem of acquiring training data pairs, [39, 33] use images with synthetically created text [6] and their corresponding originals without any text. However, the process of synthetically creating text in an image does not consider the semantic context of the image; therefore, the generated images look unreal in most cases, which might introduce unwanted bias in the training data and the trained model might not transfer well onto real-world images. In contrast, our method does not require any training data pairs of original images and their corresponding ground-truth images with the text removed. Only text region bounding box annotation is needed so that a scene text detector can be trained, but acquiring bounding box annotation is much easier than manually erasing text to generate ground-truth images.

A comparison of the features of our proposed method and methods from related works is summarized in Table 1. As we focus on erasing scene text, we compare our method only with other methods for erasing text from natural scene images.

2.2. Scene Text Detection

Early successful works on detecting text instances in natural scene images used handcrafted features such as stroke width transform [4], maximally stable extremal regions [28], and histogram of oriented gradients [34]. After the breakthrough of deep convolutional neural networks (CNNs) in computer vision tasks in recent years, first works adopting neural network approach combined handcrafted features with CNNs [10, 32]. Most recent methods follow the trends of general object detection and image segmentation and use a single neural network. A line of works inspired by object detection methods employs bounding box

	GAN	Selective erasing	Fully automatic	Training w/o image pairs
Nakamura et al. [26]	x	x	✓	x
EnsNet [39]	✓	x	✓	x
MTRNet [33]	✓	✓	partially	x
Our method	✓	✓	✓	✓

Table 1: A summarized comparison of the features of our method and related works. Note that selective erasing in our case requires human interaction, but the user can simply select from candidate regions instead of drawing them as in MTRNet [33]. When selective erasing is not required, our method is fully automatic. MTRNet can also run in a fully-automatic mode but with very limited performance.

regression. Several of these methods perform two-stage detection, incorporating a region proposal network in the model [24, 13]. Other methods use a single-stage network to carry out bounding box regression [20, 9]. Another line of methods utilizes image segmentation to localize text regions. Several recent works [23, 36] are inspired by instance segmentation and build upon Mask R-CNN [7]. Other methods use binary segmentation of text area and propose additional algorithms to correctly separate individual text regions [35, 44]. There are also works that combine the aforementioned approaches and use both bounding box regression and segmentation to detect text regions [42, 22].

In our model, we need to mask out detected text regions; therefore, we employ a segmentation-based scene text detector that explicitly produces segmentation maps of detected text regions.

2.3. Image Inpainting

Conventional image inpainting methods use image-level features to fill missing holes with texture from the surrounding area [3, 1]. The performance of these methods is limited and application on larger holes results in severe artifacts and noise. More advanced algorithms find patches that match with the context surrounding the holes and use them to fill in the blank space [2]. While producing a more realistic texture, it does not consider the semantics of the image and fails to plausibly regenerate structure of objects.

In recent years, a majority of methods for image inpainting use CNNs [29, 11, 37, 21, 40]. An image with holes is passed into a network which infers the missing pixels and outputs an image with filled in holes. CNNs have large modeling capacity which allows them to learn complex image features and they can also learn their semantics, which leads to more realistic results. Pathak et al. [29] were the first to adopt a GAN for image inpainting, which enabled them to fill large holes. To achieve both more globally and locally coherent results, the utilization of global and local discriminators [11] was proposed. Since previous research focused on rectangular holes, partial convolution [21] and gated convolution [37] were introduced to deal with free-form holes by masking or gating the responses from convolutional filters to utilize only valid image pixels.

There are countless ways to fill in missing pixel regions in an image and create a visually realistic result. Unlike other methods that are deterministic and generate only one result per an image, the model proposed by Zheng et al. [40] has an element of stochasticity, which enables generation of multiple different results per one image. Adding stochasticity to the model does not achieve only diversity in the results, but also improves their visual quality.

3. Methodology

Our proposed model consists of two separately trained modules:

1. scene text detector
2. generative adversarial network for inpainting

At inference time, the two modules are connected into one model for scene text erasing whose overall structure is illustrated in Figure 2.

3.1. Scene Text Detector

We use PSENet [35] for detecting text in natural scene images to produce text region masks in our model. It is an encoder-decoder network that outputs several segmentation maps for an image, each of which corresponds to kernels produced by shrinking the original text bounding boxes with various scales. For final bounding box detection, progressive scale expansion algorithm is employed. However, we only need segmentation maps for our scene text eraser so we extract the full-scale segmentation map produced by the neural network and use it as a mask for the inpainting module as can be seen in Figure 2.

The scene text detector is trained on the MLT dataset of multi-language scene text images [27] and ICDAR 2015 scene text dataset [15]. We experiment with both ResNet-50 and ResNet-152 [8] as the backbone networks for the model.

3.2. Inpainting Module

The model proposed by Zheng et al. [40] serves as the inpainting module in our eraser. It is a GAN that consists of

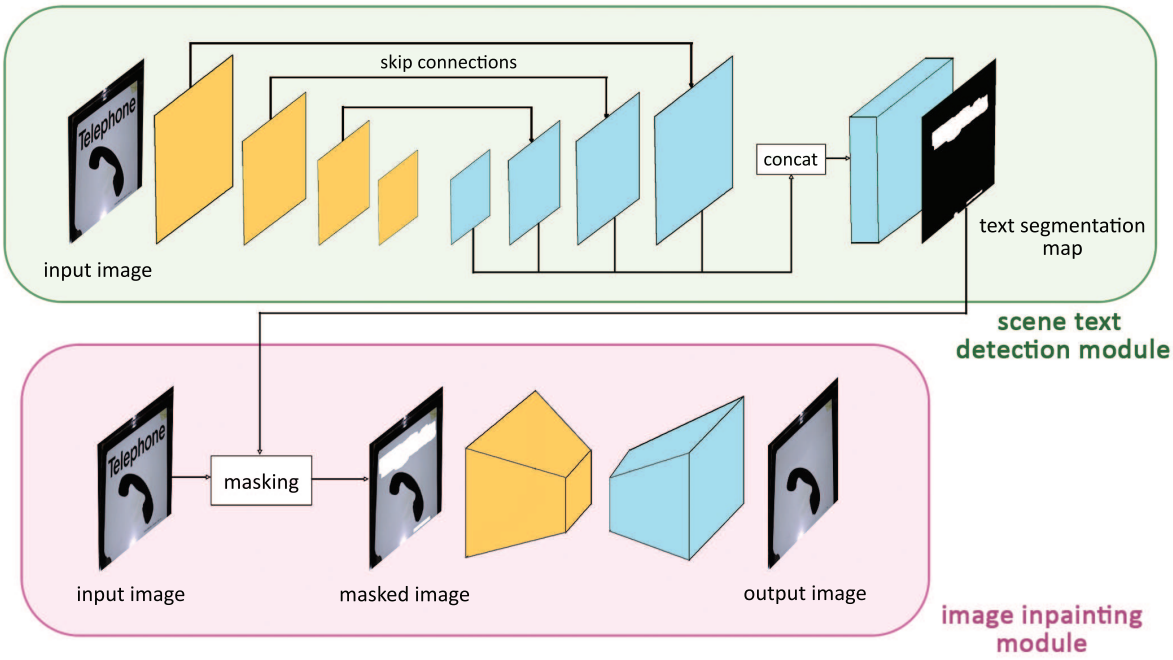

Figure 2: The overall structure of our scene text eraser model at inference time. It consists of a scene text detection module (top) that produces a segmentation map, and an inpainting module (bottom) that produces the final result. The orange and blue blocks represent encoding and decoding layers, respectively.

two parallel pipelines, both of which have a generator and a discriminator. The generative pipeline takes a masked out image as input, and the complement of the image is passed as input to the reconstructive pipeline. During training, the information extracted from the generative pipeline is passed to the reconstructive pipeline so that it can learn to restore the original image. The generative pipeline only learns to infer and fill in the missing regions from the visible parts of the image. During testing, only the generative pipeline is used.

The inpainting network is trained using one of the datasets described in Section 4.1. We exploit the size and image diversity of those datasets to obtain a robust inpainting model, necessary to produce natural-looking results.

3.3. Scene Text Eraser

To erase text from images, we combine the two modules described above into one model at test time. An input image I containing text instances is first processed by the scene text detector, which produces a binary segmentation map M of predicted text regions. The segmentation map is then used as a mask for the inpainting network.

The inpainting network takes the same image I that was processed by the scene text detector as input. At training time, a random mask is used to mask out parts of the input image, but at test time we use the segmentation map M predicted by our scene text detector to mask out text regions in the input image. The image with masked out text regions is then processed by the generator from the inpainting network, which fills in the blank regions and produces a final image with where the text is erased and missing parts are ”re-painted” in a way that they naturally blend with the background.

We do not directly train the model to learn to erase text from scene images because our method does not use pairs of original and ground-truth images for training.

The structure of the entire model is shown in Figure 2.

4. Experiments

4.1. Datasets

4.1.1 Scene Text Detection

The scene text detector is trained on MLT 2017 and ICDAR 2015 datasets. **MLT 2017** [27] is a multilingual scene text dataset consisting of 7200 training images, 1800 validation, and 9000 test images, and their corresponding annotations. **ICDAR 2015** [15] comprises 1000 annotated training images and 500 test images containing incidental text in the Latin alphabet.

4.1.2 Image Inpainting

We use inpainting modules trained on the following datasets:

- **Places2** [41] - 1.8 million images from 365 categories of indoor and outdoor scenes.
- **Paris Street View** [29] - about 15 thousand images of outdoor scenes from the streets of Paris collected from Google Street View.
- **ImageNet-1k** [31] - approximately 1.2 million training images of 1000 object categories.

4.1.3 Evaluation Datasets

ICDAR 2013 scene text dataset [16] contains 229 images for training and 233 for testing. The images were taken with a focus on the text instances. We use the test set of this dataset to evaluate the results of text erasing in terms of recall, that is how much of the text in original images is detected in the images generated by our scene text eraser.

Synthetic dataset for scene text removal proposed in [39] contains 8000 training and 800 testing image pairs. An image pair consists of an image with text synthetically placed over it and its ground-truth image that does not contain any text as shown in Figure 3. We only use the test set to evaluate the performance of our method. Note that most of the images from the test set are also included in the training set; therefore, the dataset does not necessarily test the generalization abilities of a model when used for training.

Real-world dataset for scene text removal created by Zhang et al. [39] consists of 1000 pairs of real images with scene text and their corresponding images with the text manually removed using a photo editing software as can be seen in Figure 3. The images were taken from a subset of images with English text from the MLT 2017 dataset.

4.2. Baseline for Inpainting

We set up two simple methods as baselines to compare the inpainting module of our proposed method with.

- **Maskout**. Text regions detected using a trained scene text detector are masked out from the image. This method erases all of the detected text from an image but produces visually unpleasant results by leaving holes in the image, thus severely degrading the image quality.
- **Blur**. Instead of masking out the detected text regions, we blur them by applying a Gaussian filter. Stronger filters are necessary to blur larger text regions. Therefore, we empirically find filter parameters that optimize the trade-off between visual quality and text detection recall.

Figure 3: Examples of image pairs from the synthetic dataset (top) and the real-world dataset (bottom) which were introduced in [39].

4.3. Evaluation

Quantitative results. For quantitative evaluation of our proposed method, we follow [26, 39, 33] and use an auxiliary scene text detector to investigate how much text can be detected in images after processing them through our scene text eraser. To make the comparison as fair as possible, we use the same scene text detection model [42] that was used in previous works. The performance is evaluated by computing the recall of text detection on images from the ICDAR 2013 dataset [16] in which the text was erased by our model. Recall serves as an indicator of how much of the original text was detected, so the lower the recall is, the more text instances are successfully erased by the model.

The results in Table 2 indicate that the choice of training dataset for an inpainting module does not have much influence on text detection recall. The maskout baseline yields the lowest recall since all of the detected text pixels are removed without being replaced by inpainting. However, the proposed method achieves similar text detection recall while producing visually better images.

A comparison with previous works can be seen in Table 3. MTRNet [33] achieves the lowest recall but only when ground-truth masks of text regions are used in network input. Our method yields the lowest recall when ground-truth masks are not used.

Visual quality. We also evaluate our method in terms of visual quality. For that we resort to the datasets from [39] and use the structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR) as metrics to calculate how similar the processed images with erased text are to their

ResNet-50	ICDAR 2013	Synthetic dataset		Real-world dataset	
	Recall	SSIM	PSNR	SSIM	PSNR
Maskout	**2.19**	84.03	36.06	83.57	35.17
Blur	2.49	90.69	36.06	89.78	35.20
Proposed method (Places2)	2.47	**93.69**	37.44	**92.53**	36.41
Proposed method (Paris)	3.00	92.53	37.15	91.52	36.19
Proposed method (ImageNet)	2.56	93.58	**37.55**	92.32	**36.51**

ResNet-152	ICDAR 2013	Synthetic dataset		Real-world dataset	
	Recall	SSIM	PSNR	SSIM	PSNR
Maskout	**0.55**	83.94	36.03	83.44	35.21
Blur	0.67	90.51	36.02	89.63	35.23
Proposed method (Places2)	0.64	**93.64**	37.46	**92.73**	36.52
Proposed method (Paris)	1.26	92.56	37.16	91.67	36.27
Proposed method (ImageNet)	0.82	93.60	**37.62**	92.39	**36.59**

Table 2: Performance comparison of baselines and the proposed method. We use three different datasets for evaluation of scene text erasing in terms of text detection recall, and visual quality of generated results. Lower recall and higher SSIM and PSNR indicate better performance. The top table shows results with ResNet-50 as the backbone network for the scene text detector, the bottom table shows results using ResNet-152. The names in the parentheses specify which dataset the inpainting module was trained on.

	ICDAR 2013
	Recall
Nakamura et al. [26]	10.08
EnsNet [39]	5.66
MTRNet (wo/ GT mask) [33]	29.11
MTRNet (w/ GT mask) [33]	**0.18**
Ours (ResNet-50)	2.47
Ours (ResNet-152)	**0.64**

Table 3: Comparison of performance of our proposed method and previous works on the auxiliary task of detecting text in images processed by a scene text eraser. Recall is an indicator of how much of the original text is detected. Lower recall means that more text has been successfully erased.

	Synthetic dataset [39]	
	SSIM	PSNR
EnsNet [39]	**96.44**	37.36
Ours (ResNet-50)	93.69	37.44
Ours (ResNet-152)	93.64	**37.46**

Table 4: Comparison of visual quality of images generated by our proposed method and related work. Higher SSIM and PSNR values imply better performance. Our method does not yield as good results as [39]; however, it is not trained on synthetic data and does not use original and ground-truth image pairs for training. Here we state our results with Places2 dataset used for training.

ground-truth counterparts without text. Higher SSIM and PSNR values imply better performance.

Table 2 shows that the proposed method achieves notably better results than the baseline regardless of which dataset is used in the training of the inpainting module. In terms of both metrics, inpainting modules trained on the ImageNet-1k and Places2 datasets produce similar results. The inpainting module trained on the Paris Street View dataset renders slightly worse results which confirms that a large amount of data is highly beneficial for increasing the performance.

We also compare the performance with previously reported results in Table 4. Our method yields a lower SSIM, but it is not trained on the synthetic dataset. In contrast, EnsNet [39] uses the dataset for training. Our model expects realistic input, whereas the synthetic images considerably deviate from reality. In particular, text does not stretch over multiple semantically different regions in real scenes (e.g., a foreground object and background), but the synthetic dataset contains many such examples. Also note that the synthetic dataset introduced in [39] contains test images in the training data, which gives any method that is trained on the dataset a significant advantage.

Qualitative results. Examples of scene text erasing results can be seen in Figure 4. Our proposed method can remove the text from images and fill in the missing pixels in a way that makes them blend with the surrounding area. Unlike baseline methods, it produces images that can give the impression that there never was any text in the image. Furthermore, it completely removes the text pixels unlike

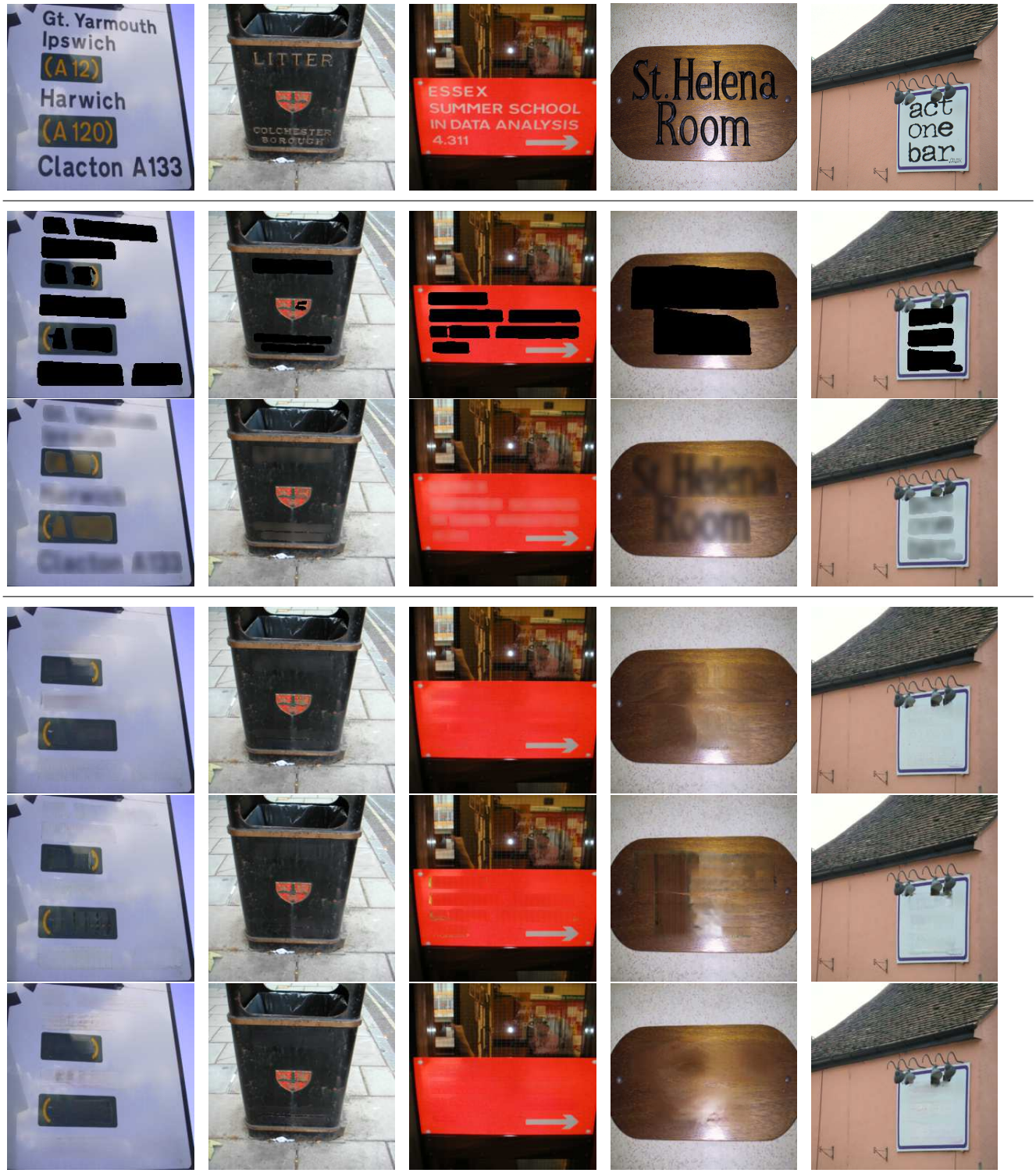

Figure 4: Examples of scene text erasing on images from the ICDAR 2013 dataset. The top row shows the original images containing text instances. The second and third row show results produced by baselines - maskout and blur, respectively. Results of our method can be seen in rows 4 to 6, where each row shows results produced with inpainting modules trained on different datasets: Places2, Paris Street View, and ImageNet-1k (in this order).

Figure 5: Examples of two types of failure cases: unsuccessful text erasing caused by imperfect text detection (left), a failure to fill the masked out region in a way that would blend with the surrounding area (right).

blurring, which is prone to producing results that a state-of-the-art method for scene text reading or a human could read in some cases.

Images produced using an inpainting module trained on the Paris Street View dataset contain more artifacts than those generated by an inpainting module trained on Places2 or ImageNet-1k. This is likely caused by the difference in the size of the datasets. Paris Street View is approximately 100 times smaller than the other datasets so it contains much less information which the network can learn from. Inpainting modules trained on Places2 and ImageNet-1k generate visually similar results, which coincides with the qualitative results.

Figure 5 illustrates two types of failures that can occur. The first column shows a case of unsuccessful text erasing caused by imperfect detection of text by the scene text detector. The second column demonstrates an example where text was successfully detected but the inpainting network was not able to fill in the masked out region in a way that would blend with the surrounding area and look natural. Imperfect text detection is likely to occur when the appearance of the text is unusual or the text is very small. On the other hand, producing good inpainting results is difficult when the text region is large or when the background contains complex patterns.

5. Conclusion

We have proposed a method for removing text from natural scene images that does not require training data pairs of original images containing text instances and their corresponding ground-truth images with the text removed. We bypassed the expensive process of creating ground-truth data by approaching the problem of erasing text from images as a general inpainting task and combined it with scene text detection.

The results show that our method can remove text from images and fill in the created blank space to naturally blend with the background. The qualitative and quantitative evaluation indicates that masking out detected text and using inpainting to fill in the holes is a safer and visually better way to conceal scene text than simply blurring detected text and that our method produces results that are competitive with existing methods.

We made attempts to further improve the performance by finetuning the scene text detection and inpainting modules together in an end-to-end manner but we were not able to achieve better results. Improving the performance by training in end-to-end manner thus remains future work.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP19K22861.

References

[1] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, 2001.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, volume 28, 2009.

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 2000.

[4] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR*, 2010.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[6] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, 2016.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[9] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li. Single shot text detector with regional attention. In *CVPR*, 2017.

[10] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *ECCV*, 2014.

[11] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4), 2017.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[13] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo. R2cnn: Rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017.

[14] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[15] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. Icdar 2015 competition on robust reading. In *ICDAR*, 2015.

[16] 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 *ICDAR*, 2013.

[17] M. Khodadadi and A. Behrad. Text localization, extraction and inpainting in color images. In *20th Iranian Conference on Electrical Engineering (ICEE2012)*, pages 1035–1040. IEEE, 2012.

[18] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon. Deep blind video decaptioning by temporal aggregation and recurrence. In *CVPR*, 2019.

[19] C. W. Lee, K. Jung, and H. J. Kim. Automatic text detection and removal in video sequences. *Pattern Recognition Letters*, 24(15):2607–2623, 2003.

[20] M. Liao, B. Shi, and X. Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE Transactions on Image Processing*, 27(8):3676–3690, 2018.

[21] 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 *ECCV*, 2018.

[22] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan. Fots: Fast oriented text spotting with a unified network. In *CVPR*, 2018.

[23] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *ECCV*, 2018.

[24] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018.

[25] A. Mosleh, N. Bouguila, and A. B. Hamza. Automatic inpainting scheme for video text detection and removal. *IEEE Transactions on image processing*, 22(11):4460–4472, 2013.

[26] T. Nakamura, A. Zhu, K. Yanai, and S. Uchida. Scene text eraser. In *ICDAR*, 2017.

[27] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *ICDAR*, 2017.

[28] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *CVPR*, 2012.

[29] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[30] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[32] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. Lim Tan. Text flow: A unified text detection system in natural scene images. In *ICCV*, 2015.

[33] O. Tursun, R. Zeng, S. Denman, S. Sivipalan, S. Sridharan, and C. Fookes. Mtrnet: A generic scene text eraser. *ICDAR*, 2019.

[34] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *ICCV*, 2011.

[35] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao. Shape robust text detection with progressive scale expansion network. In *CVPR*, 2019.

[36] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li. Scene text detection with supervised pyramid context network. *arXiv preprint arXiv:1811.08605*, 2018.

[37] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018.

[38] H. Zhang, V. Sindagi, and V. M. Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017.

[39] S. Zhang, Y. Liu, L. Jin, Y. Huang, and S. Lai. Ensnet: Ensconce text in the wild. In *AAAI*, 2019.

[40] C. Zheng, T.-J. Cham, and J. Cai. Pluralistic image completion. *CVPR*, 2019.

[41] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 2017.

[42] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: an efficient and accurate scene text detector. In *CVPR*, 2017.

[43] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[44] Y. Zhu and J. Du. Textmountain: Accurate scene text detection via instance segmentation. *arXiv preprint arXiv:1811.12786*, 2018.