Scene Text Editing

[1]*Felix Liawi* (潘建琿) [1]*Chiou-Shann Fuh* (傅楸善)

[1]Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan,
*E-mail: r11922177@ntu.edu.tw fuh@csie.ntu.edu.tw

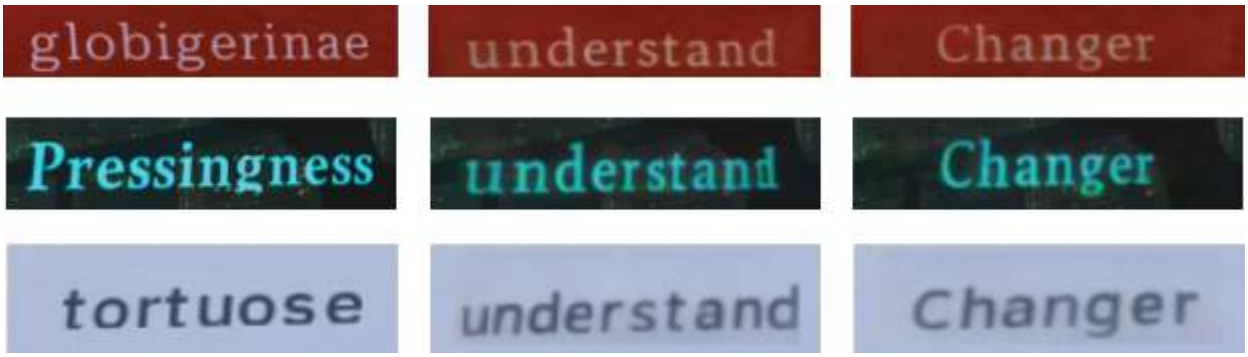

Fig. 1. Text scene image text modifications. The left image represents the original text scene, the center image depicts the text scene after the application of the new text "understand," and the right image illustrates the text scene after the application of the new text "Changer".

ABSTRACT

Scene Text Editing (STE) aims to substitute text in an image with new desired text while preserving the background and styles of the original text. However, current methods face challenges in generating edited text images that are clear and legible, mainly due to the diverse text types and complex background textures. In this paper, we introduce a three-stage framework for transferring texts across text images.

Firstly, we propose a text swapping network designed specifically for replacing original text with the new text. Secondly, we apply a background inpainting network that learns to reconstruct the background images to our scene text editing. The purpose of this network is to fill in the regions where the original text has been removed, ensuring a visually consistent and coherent background. Finally, we employ a fusion network to combine the outputs of the text swapping network and the background inpainting network, resulting in the generation of the final edited image.

By utilizing our proposed method, we enable the modification of text in input images while preserving the overall visual integrity of the scene. Our approach addresses the limitations of existing techniques and demonstrates the potential for generating high-quality and visually plausible edited text images. Figure 1 shows the result of text scene image modifications.

Keywords: Scene Text Editing, Text Replacement, Image Text Transfer

1. INTRODUCTION

Editing text in an image while preserving its style and background presents a challenging computer vision task. Manual editing by humans can be a time-consuming process, often taking hours to edit a single image. This

task, known as Scene Text Editing (STE), has garnered significant attention due to its potential applications in various domains, such as creating advertisements, magazines, and gaming scenes.

The objective of this work is to develop an algorithm capable of automatically changing the text in an image, thereby streamlining the editing process. The automation of STE can greatly enhance efficiency and productivity in industries that heavily rely on visual content creation such as advertising, etc. By enabling automated text editing, professionals and designers can focus on higher-level tasks, leading to faster turnaround times and increased creativity in their work.

Additionally, STE is closely related to other tasks in the field of computer vision, such as scene text detection and scene text recognition. Previous studies have demonstrated that deep neural networks perform better when trained on larger volumes of data. However, manually collecting enormous datasets for these tasks can be resource-intensive and time-consuming. By leveraging STE techniques, the task of gathering a vast scene dataset for training purposes becomes more feasible. This opens up new avenues for improving the performance of scene text detection and recognition systems through the use of larger and more diverse training datasets.

The potential impact of automating STE extends beyond the realm of content creation. It has the potential to revolutionize industries that heavily rely on text-based information, such as publishing, advertising, and branding. By efficiently modifying text in images, businesses can quickly adapt their visual materials to cater to changing requirements, target different demographics, or experiment with various design options.

In this paper, we propose a novel algorithmic framework for STE that combines advancements in deep learning, image processing, and text replacement. Our approach aims to address the limitations of manual editing while preserving the style and background of the original image. We conduct extensive experiments to evaluate the effectiveness and efficiency of our method, showcasing its potential for practical applications in real-world scenarios.

2. RELATED WORK

2.1. Style Transfer

Style transfer is a challenging task that involves the transformation of the visual style of a source image to a target image. The majority of existing methods utilize encoder-decoder architectures to generate the target image. Luan et al. (2017) [2] introduced a deep photo style transfer technique that employs a mapping network to learn a mapping from an input photo to the parameters of a synthesis network, which then generates the stylized output image from a random noise vector. Li et al. (2017) [3] proposed a texture synthesis method based on feed-forward neural networks that trains a generator network to perform style transfer by adjusting the mean and variance of feature maps, and introduces a diversity regularization term to encourage the generation of a wider range of textures. Karras et al. (2019) [4] implemented a mapping network that learns to map vector noise z to weights, which are then fed to a synthetic network to create the target image.

2.2. Text Image Synthesis

Text image synthesis has become an increasingly important technique for enhancing the performance of deep neural networks in various text-related tasks, such as data augmentation for text detection and recognition. For example, Krishnan, Praveen et al. (2016) [5] proposed a method for Generating Synthetic Data for Text Recognition, which involves synthesizing text images using a variety of fonts and styles, followed by realistic augmentation techniques such as rotation, scaling, and blurring. They demonstrate that combining the synthetic data with real data leads to improved recognition

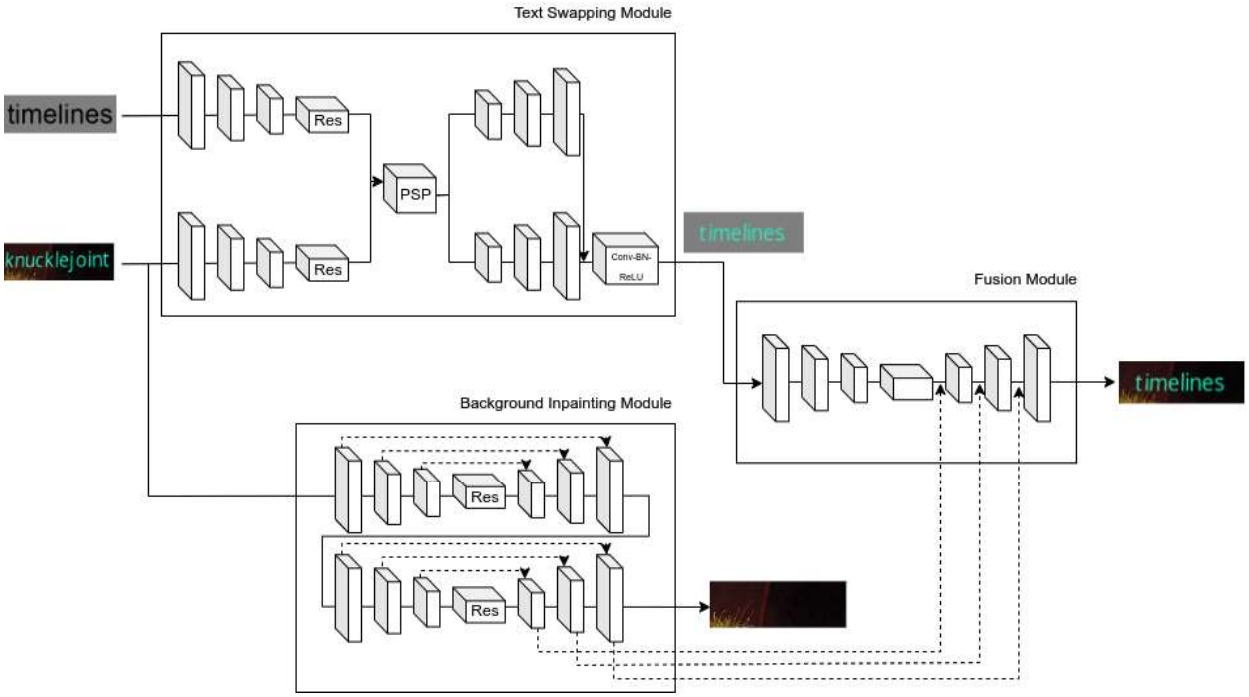

Fig. 2. The overall pipeline of proposed method. Consists of Background Inpainting Module, Text Swapping Module and Fusion Module.

performance on multiple benchmarks, highlighting the potential of synthetic data as a valuable resource for training text recognition models. Yim, Moonbin, et al. (2021) [6] proposed a synthetic text image generator called SynthTIGER, which utilizes a Generative Adversarial Network (GAN) to generate synthetic text images, and employs a novel embedding-based technique to condition the GAN to generate diverse and realistic images.

2.3. Scene Text Editing

Scene text editing presents a formidable challenge as it strives to transfer text style while faithfully preserving the background in a realistic manner. Numerous recent studies in this domain have employed GAN-based models. For instance, Wu et al. (2019) [8] propose a hierarchical editing approach, consisting of three sub-networks: background inpainting, text swapping, and a fusion network. Conversely, Roy et al. (2020) [7] introduce a Font Adaptive Neural Network, albeit with constraints limited to character-level modifications. Notably, Qu et al. (2022) [11] put forth an innovative network, named MOSTEL (MOdifying Scene Text Image at strokE Level), that modifies scene text images at the stroke level.

3. METHODOLOGY

The proposed methodology aims to generate highly realistic results while ensuring the preservation of the background to the maximum extent possible. This objective is achieved through the utilization of a three-module network. Figure 2 illustrates the overall pipeline of the proposed method, which comprises the Background Inpainting Module (BIM), Text Swapping Module (TSM), and Fusion Module (FM).

The BIM module takes the source image as input and generates a background devoid of any text. Subsequently, the TSM module accepts both the source image and a target text image in standard format as inputs, and produces two transfer results: one in

grayscale skeleton form and another in RGB format. Finally, the FM module takes the text-free background and the RGB transfer result as inputs, and generates the final output of the scene text editing process.

Further details regarding each module will be elaborated upon in the subsequent sections.

3.1. Background Inpainting Module

The objective of the Background Inpainting Module (BIM) is to generate a text image that is free of any text artifacts. Inspired by the work of Tang et al. (2021) [9], BIM adopts a double encoder-decoder architecture.

The first encoder-decoder architecture, called the Stroke Mask Module (SMM), is utilized to extract the text mask. The second structure, known as the Text Removal Module (TRM), is responsible for removing the text from the original image using the text mask and the original image as inputs.

The SMM encoder comprises three downsampling blocks and one residual block, while the decoder consists of three upsampling blocks. On the other hand, the TRM encoder consists of three partial convolution downsampling blocks, one self-attention block, and the decoder consists of three partial convolution upsampling blocks. This architecture allows the TRM module to effectively remove the text while preserving the integrity of the original background image.

To make sure that BIM generates good results, we adopted L2 loss.

$$L_{b\_i} = \|T_b - O_b\|_2$$

3.2. Text Swapping Module

Text instances in our physical world exhibit a diverse range of shapes, encompassing both horizontal and curved forms. The primary aim of the proposed Text Swapping Module (TSM) is to effectively replace the textual content present in a given style image I_s while meticulously preserving the original styles and background. To accomplish this objective, we introduce a Text Swapping Module that facilitates the acquisition of a learned mapping between the content image and the style image. Both the style and content inputs undergo a thorough processing pipeline involving three layers of encoder-decoder networks as well as a ResNet layer. Additionally, the content features are merged with the style features employing a PSP network Zhao et al. (2017) [10], followed by the utilization of a double encoder-decoder network. One of the encoder-decoder networks focuses on generating the skeleton structure denoted as O_{ts_sk}, while the other network is responsible for generating the foreground images referred to as O_{ts_fg}.

To make sure that TSM generates good results, we adopted L2 loss between O_{ts_fg} and original image T_{ts_fg}.

$$L_{ts\_fr} = \|T_{fr} - O_{fr}\|_2$$

Additionally, we have incorporated the dice loss, as proposed by Sudre et al. (2017) [14], to further enhance our approach. This loss function is applied between the O_{ts_sk} and the original skeleton image T_{ts_s}, to improve the quality and accuracy of the generated results.

$$L_{ts\_sk} = 1 - \frac{2 \times \text{intersection}}{\text{union} + \text{intersection}}$$

The loss function utilized for this module is defined as follows

$$L_{ts} = L_{ts\_fr} + L_{ts\_sk}$$

3.3. PSP Network

To incorporate the style and content features effectively, we employ a Pyramid Scene Parsing (PSP) module. The PSP module is designed to capture multi-scale contextual information from both the style and content images. It operates by dividing the input feature maps into multiple regions and extracting features at different scales.

In our approach, the style and content features are fed into the PSP module, which

consists of parallel convolutional branches with varying receptive field sizes. Each branch captures contextual information at a different scale, enabling a comprehensive understanding of the scene. The resulting feature maps from each branch are then fused together to form a rich representation that encompasses both local and global context.

By leveraging the PSP module, our model gains the ability to effectively encode and incorporate the intricate details and contextual cues from both the style and content images. This enables the Text Swapping Module (TSM) to generate accurate and visually appealing foreground images O_{ts_fg} while preserving the desired style characteristics.

3.4. Fusion Module

The Fusion Module (FM) is designed to produce a coherent text image with a natural-looking background. This module utilizes an encoder-decoder architecture, where the encoder component comprises three downsampling blocks and one residual block. Similarly, the decoder component comprises three upsampling blocks, as depicted in Figure 2.

To enhance the realism of the generated results, the decoder layers of FM employ a concatenation strategy. Specifically, each decoder layer takes the concatenation of the decoder output and the corresponding output from the Text Removal Module (TRM) inside Background Inpainting Module (BIM) decoder layer as its input. This approach enables the decoder layers to leverage the TRM decoder layer's contextual information, thus improving the overall coherence and quality of the generated text images.

The loss function on the final result uses both GAN loss and L_2 loss.

$$L_f = \lambda_{f_1}\mathbb{E}[\log D_f(T_f, I_f) + \lambda_{f_2}\log(1 - D_f(O_f, I_t)] \cdot + \|T_f - O_f\|_2$$

To make sure that the final results are readable, we adopted a recognizer loss that uses cross-entropy loss.

$$L_{rec} = -\frac{1}{N}\sum_{i=1}^{N}\log(p_i|g_i)$$

where p_i and g_i is the prediction and ground truth of the labels. N indicates the maximum length which is the sum of lowercase and uppercase characters.

To generate a more realistic image, we adopted VGG-loss, which is divided into a perceptual loss and style loss.

$$L_{vgg} = \lambda_{v_1}L_{per} + \lambda_{v_2}L_{style}$$

The whole loss function can be expressed as

$$L = L_{b\_i} + L_{ts} + L_f + \lambda_1 L_{vgg} + \lambda_2 L_{rec}$$

4. EXPERIMENTS

4.1. Implementation Details

We adopt a similar approach as presented in Qu et al. (2022) [11] to generate synthetic pairwise images. To create a comprehensive benchmark dataset, we utilize a diverse collection of over 300 fonts and 10,000 background images. This can be generated using the publicly available SRNet-Datagen [12]. This results in a training set comprising 1 million images and a test set consisting of 5,000 images. For uniformity, the input images are resized to dimensions of 64 × 256 pixels, and a training batch size of 16 is employed.

To optimize the entire network, we employ the Adam optimizer [13] with β1 = 0.9 and β2 = 0.999. Our model implementation is carried out using the PyTorch framework [1]. The training process entails iterating for 500,000 iterations, which requires approximately 4-5 days to complete on a single Nvidia GeForce RTX 3090 GPU. This computational setup ensures efficient model training and facilitates the generation of high-quality results.

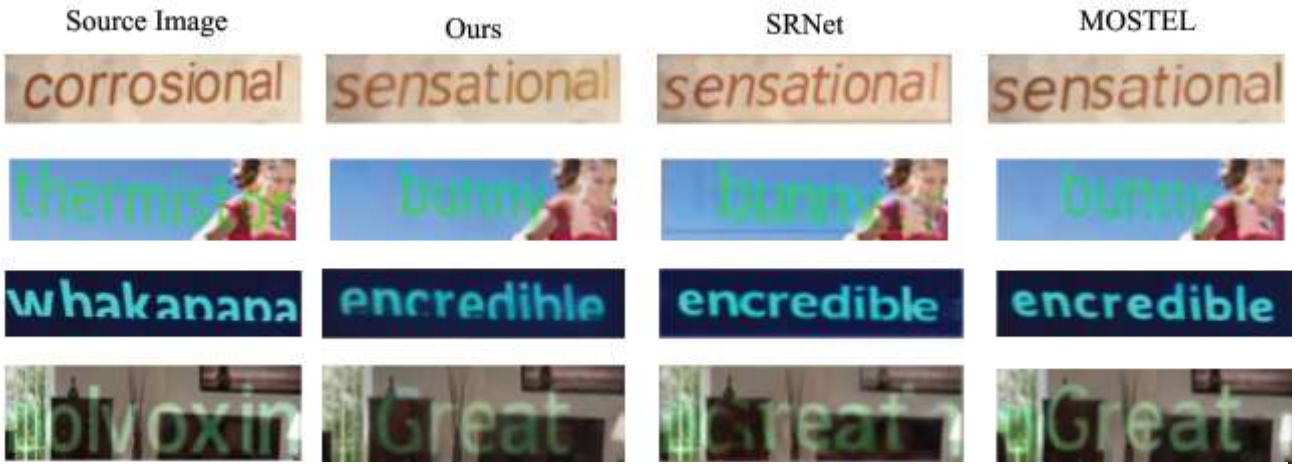

Fig. 3. Some qualitative examples of our work in comparison to the previous works SRNet and MOSTEL.

4.2. Benchmark Datasets

To assess the effectiveness of our proposed method, we conduct comprehensive evaluations on a synthetic dataset. This dataset can be generated utilizing the publicly available SRNet-Datagen [12]. The utilization of this framework allows us to create a diverse and representative dataset that is well-suited for evaluating our method's performance.

4.3. Evaluation Metrics

We adopt the commonly used metrics in scene text editing to evaluate our proposed method.

- MSE or Mean Square Error, also known as l2 error
- PSNR or Peak Signal Noise Ratio, which computes the ratio of peak signal to noise
- SSIM or Structural Similarity Index Measurement by Wang et al. (2004) [15], which computes the mean structural similarity index between two images
- FID or Fréchet Inception Distance by Heusel et al. (2017) [16], which represent the distance of image in vector representations

4.4. Comparison with Prior Work

To evaluate our proposed method, we compared it with two types of scene text editing method: SRNet proposed by Wu et al. (2019) [8], MOSTEL proposed by Qu et al. (2022) [11]. We use our generated benchmark dataset to test these two models.

4.5. Quantitative Result

In Table 1, we give some quantitative results of our method and other two competing methods. Our proposed method surpasses the other implementations in almost all metrics except FID. The average l2 error decreased by over 0.003, the average PSNR increased by over 0.1, and the average SSIM increased by over 0.05 than the second-best method.

Method	MSE↓	PSNR ↑	SSIM ↑	FID↓
SRNet	0.0154	19.548	0.676	25.832
MOSTEL	0.0122	21.049	0.719	**17.202**
Ours	**0.0098**	**22.096**	**0.763**	19.051

Lower MSE, higher PSNR, higher SSIM and lower FID represent how similar the generated result is compared to the ground truth.

4.6. Qualitative Result

In comparison to the previous works SRNet proposed by Wu et al. (2019) [8] and MOSTEL proposed by Qu et al. (2022) [11], our proposed method exhibits superior results, as demonstrated in Figure 3. Notably, our model excels in maintaining the original font style, including its unique characteristics and coloration. As evidenced in the third example of the third row, our model successfully transfers the style of cut text, a task that remains challenging for other existing methods. Furthermore, the fourth example in the same row showcases our method's proficiency in handling complex backgrounds, effectively preserving text while other approaches struggle to fully remove the original text. These examples provide compelling evidence of the enhanced capabilities and robustness of our proposed method in various challenging scenarios.

4.7. Limitation

While our method demonstrates capability in handling most scene text images, it is important to acknowledge certain limitations. Specifically, our proposed approach may encounter difficulties when attempting to modify text in scene text images that exhibit highly intricate structures or employ rare font shapes. Another issue is the occasional blurriness observed in our generated images when compared to their original counterparts and existing works. These limitations signify areas for further improvement and optimization in our methodology. Figure 4 shows some failed cases of our proposed method.

Fig. 4. Failure cases.

5. CONCLUSION AND FUTURE WORK

In this study, we have presented an end-to-end trainable deep neural network for scene text editing. Our proposed model enables the replacement of text in scene text images while preserving the original style, including background texture and font style. By dividing our deep neural network into three subnetworks, we have achieved a model that surpasses the performance of existing benchmark metrics.

Through extensive experimentation and evaluation, we have demonstrated the effectiveness of our approach in accurately editing scene text while maintaining the visual coherency of the image. The results indicate that our model is capable of handling various scene text editing tasks with great accuracy and fidelity.

Looking ahead, there are several avenues for further research and development in the field of scene text editing. The utmost priority lies in resolving the limitations mentioned earlier to enhance our proposed method. Another important direction is addressing more complex scenarios, such as images with noisier backgrounds or unconventional font styles. These challenges require robust algorithms that can handle diverse text appearances and backgrounds, enhancing the versatility and applicability of scene text editing.

Furthermore, exploring style transfer across different languages and specific font styles, like signatures, presents an intriguing avenue for future work. Transferring font style between languages would facilitate multilingual scene text editing, enabling users

to seamlessly modify text in images irrespective of the source language.

Additionally, by learning how to transfer specialized font styles, such as signatures, can help to improve document forgery detection or artistic text manipulation.

6. REFERENCES

[1] A. Paszke, et al. "Pytorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems,* https://arxiv.org/pdf/1912.01703.pdf, 32 pp. 1-12, 2019.

[2] F. J. Luan, S. Paris, E. Shechtman, and K. Bala. "Deep Photo Style Transfer," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, pp. 4990-4998, 2017.

[3] L. Y. Jun, C. Fang, Y. J. Mei, W. Z. Wen, L. Xin, and Y. M. Hsuan. "Diversified Texture Synthesis with Feed-Forward Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, pp. 3920-3928. 2017.

[4] T. Karras, S. Laine and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 4396-4405, doi: 10.1109/CVPR.2019.00453.

[5] Krishnan, Praveen, and C. V. Jawahar. "Generating Synthetic Data for Text Recognition." *arXiv preprint arXiv:1608.04224* https://arxiv.org/pdf/1608.04224.pdf pp. 1-5, 2016.

[6] M. B. Yim, et al. "SynthTIGER: Synthetic Text Image GEneratoR towards Better Text Recognition Models," *Proceedings of International Conference on Document Analysis and Recognition,* Lausanne, Switzerland, 16, pp. 109–124, 2021.

[7] P. Roy, S. Bhattacharya, S. Ghosh and U. Pal, "STEFANN: Scene Text Editor Using Font Adaptive Neural Network," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020 pp. 13225-13234.

[8] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. 2019. Editing Text in the Wild. *In Proceedings of the 27th ACM International Conference on Multimedia (MM '19).* Association for Computing Machinery, New York, NY, USA, 1500–1508. https://doi.org/10.1145/3343031.3350929

[9] Z. Tang, T. Miyazaki, Y. Sugaya and S. Omachi, "Stroke-Based Scene Text Erasing Using Synthetic Data for Training," in *IEEE Transactions on Image Processing*, vol. 30, pp. 9306-9320, 2021, doi: 10.1109/TIP.2021.3125260.

[10] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, "Pyramid Scene Parsing Network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Honolulu, HI, USA, 2017, pp. 6230-6239, doi: 10.1109/CVPR.2017.660.

[11] Y. D. Qu, Q. F. Tan, H. T. Xie, J. J. Xu, Y. X. Wang, and Y. D. Zhang. "Exploring Stroke-Level Modifications for Scene Text Editing." *arXiv preprint arXiv:2212.01982* https://arxiv.org/pdf/2212.01982.pdf pp.1-9, 2022.

[12] Youdao-ai. (2020). SRNet-Datagen. GitHub. https://github.com/youdao-ai/SRNet-Datagen

[13] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, Banff, AB, Canada, 2018, pp. 1-2, doi: 10.1109/IWQoS.2018.8624183.

[14] C. H. Sudre, W. Q. Li, T. Vercauteren, S. Ourselin, M. J. Cardoso. "Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations." *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*. Springer International Publishing, 2017.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, April 2004, doi: 10.1109/TIP.2003.819861.

[16] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium." *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, December 2017, pp. 6629-6640.