FASTER: A Font-Agnostic Scene Text Editing and Rendering framework

Alloy Das [1] * Sanket Biswas[2] * Prasun Roy[3] Subhankar Ghosh[3] Umapada Pal[1] Michael Blumenstein[3]

Josep Lladós[2] Saumik Bhattacharya[4]

[1]CVPRU, Indian Statistical Institute, Kolkata [2]CVC, Universitat Autònoma de Barcelona
[3]FEIT, University of Technology Sydney, Australia [4]ECE, Indian Institute of Technology, Kharagpur

Abstract

Scene Text Editing (STE) is a challenging research problem, that primarily aims towards modifying existing texts in an image while preserving the background and the font style of the original text. Despite its utility in numerous real-world applications, existing style-transfer-based approaches have shown sub-par editing performance due to (1) complex image backgrounds, (2) diverse font attributes, and (3) varying word lengths within the text. To address such limitations, in this paper, we propose a novel ***font-agnostic scene text editing and rendering*** *framework, named* ***FASTER****, for simultaneously generating text in arbitrary styles and locations while preserving a natural and realistic appearance and structure. A combined fusion of target mask generation and style transfer units, with a cascaded self-attention mechanism has been proposed to focus on multi-level text region edits to handle varying word lengths. Extensive evaluation on a real-world database with further subjective human evaluation study indicates the superiority of FASTER in both scene text editing and rendering tasks, in terms of model performance and efficiency. Our code will be released upon acceptance.*

*These authors have contributed equally to the work.

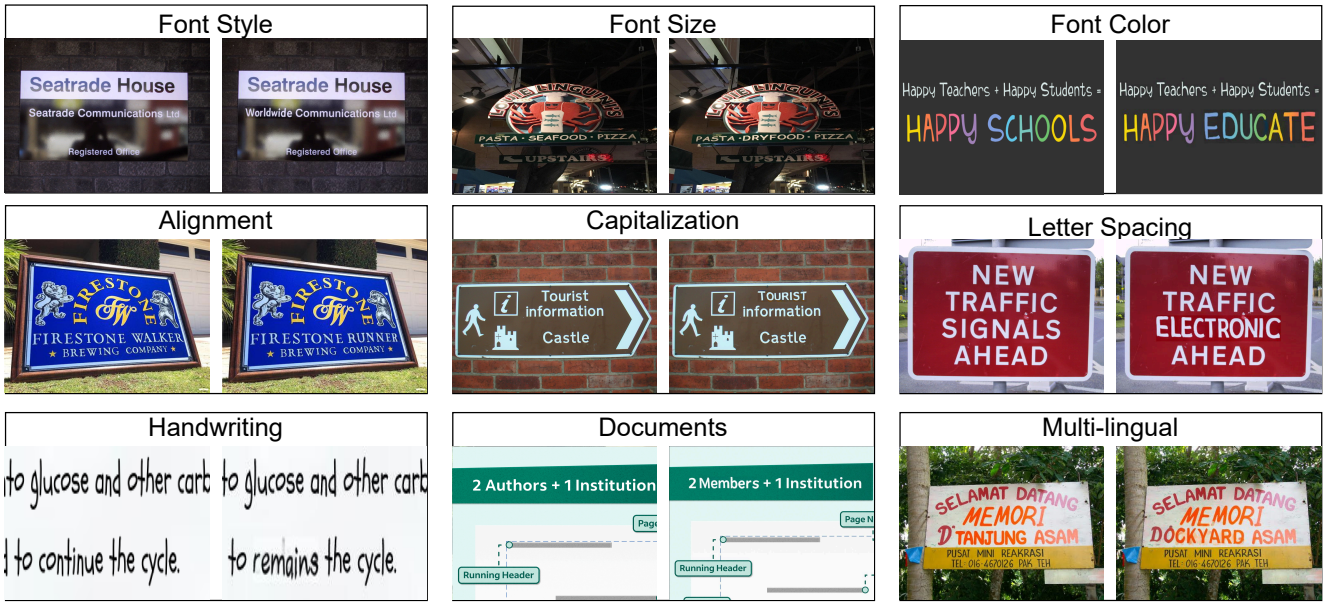

Figure 1. Given an **image** and the **desired text** to render, FASTER performs appropriate edits on the target text regions in complex real scenes with high consistency and realism on a wide range of typefaces and multiple font attributes with varying word lengths.

1. Introduction

In user interfaces, advertisements, graphic design, and augmented reality applications, ensuring a pleasing and immersive user experience is paramount. Text is a fundamental component within visual compositions, anchoring meaning, guiding attention and shaping a document image's overall *visual coherence* by adhering to attributes such as font style, size, color, and alignment [6]. In recent years, there has been a surging interest in the field of Scene Text Editing (STE) due to its practical applications across multiple tasks which include text image synthesis [31, 43, 64], text style transfer [1, 2, 70], and augmented reality translation [7, 15, 16]. Prior approaches [36, 41, 44, 49, 58, 60] have essentially formulated it as a style transfer problem using generative adversarial networks (GANs) [18]. Yet, the pivotal questions that remain unanswered are: 1) How well can STE models execute seamless text editing while retaining various text attributes, such as font style, size, color, alignment, case distinction, and letter spacing, in complex real-world scenes? 2) How does the inference time of these models impact their practical usefulness and efficacy in real-world applications, without compromising the text editing and rendering quality?

Alternative approaches in STE [22, 51, 58] often use a reference image to guide text attribute rendering. However, these methods struggle with diverse font attributes and arbitrary geometric transformations due to content and style entanglement. Diffusion-based approaches [8, 28] generate high-quality text images but lack precise control over font style, placement, and complex backgrounds. This is due to (1) **Limited Adaptability and Control**: Focusing on high-level text descriptions [9] without capturing contextual backgrounds limits adaptability. (2) **Computational Complexity**: Diffusion methods are computationally intensive, hindering real-time editing. (3) **Lack of Positional Guidance**: Text rendering requires alignment, spacing, and formatting considerations not handled by diffusion models.

In this work, we propose FASTER, a novel two-stage GAN-based text style transfer architecture. In Stage-I, a style translation block (STB) generates a target style mask. In Stage-II, a content translation block (CTB) conditions text image attributes to generate content over the Stage-I output. This top-down approach improves upon existing designs [58, 60] that often suffer from global information loss and degraded output quality. Unlike TextStyleBrush [34], which relies heavily on a pre-trained font classifier and text recognizer and struggles with background attributes in complex images, FASTER incorporates a simple and efficient U-Net module [48] to preserve the original text appearance by affecting only non-text regions. Additionally, we introduce a cascaded self-attention module in the Stage-I decoder, inspired by prior works [40, 56, 67], to model hierarchical, multi-level dependencies across text regions. This decomposition into multiple cascaded layers with sigmoid attention units allows FASTER to dynamically prioritize relevant text parts, enabling faster and more accurate text editing and rendering. Figure 1 illustrates some qualitative results on some complex real-world examples that demonstrate how the precise edits on the target text preserve font (or typeface) characteristics of the text and its layout composition relative to the foreground style. To further support with quantitative evidence, our proposed FASTER design shows a substantial *improvement of* ***-23.83*** *in perceptual FID* over second-best SwapText [60] and a notable enhancement in *inference time* (measured in milliseconds (ms) per image) compared to the runner-up SRNet [58], achieving a *cost reduction of* ***15.75*** *ms*. The major contributions and novelties of this work can be divided into four folds: (1) FASTER is a novel STE framework using a *two-staged style-transfer methodology with positional guidance* for robust text attribute conditioning. (2) The proposed framework integrates a *cascaded self-attention module* in the decoder for efficient scaling and pre-training, enhancing *real-world text-editing capabilities* (e.g., handwritten fonts, documents, multi-lingual scripts) and achieving the fastest inference speed. (3) We also adapt a *novel combination of learning objectives* for the GAN generator which allows high-quality text image rendering without sacrificing model efficiency. (4) A comprehensive evaluation toolkit based on [44] and human evaluation study has been provided to ensure thorough quantitative and subjective assessments of text editing and rendering quality.

2. Related Work

Text Image Synthesis: Image synthesis and rendering have been extensively studied in computer graphics [13]. Text image synthesis serves as a data augmentation technique for text identification and detection. Gupta et al. [19] developed an engine to produce synthetic text images, while Jaderberg et al. [27] created a word generator for synthetic word images which had immense influence in domain-generalized scene text spotting tasks [10–12]. The aim of text image synthesis is to insert text into semantically significant areas of a background image. However, factors such as font size, perspective, and illumination affect the realism of synthesized text images. To address this, Zhan et al. [64] integrated semantic coherence, visual attention, and adaptive text presentation, achieving visually accurate but still distinguishable synthetic images with limited font options. Recently, Biswas et al. [5] proposed an approach to generate synthetic documents with relevant text content with variable fonts.

To address these restrictions, recent studies have explored GAN-based picture synthesis algorithms. Zhan et al. [66] introduced a spatial fusion GAN that combines a geometry synthesizer with an appearance synthesizer for realistic synthesis in both geometry and appearance spaces. Yang et al.'s [63] framework allows control over glyph styles using an adjustable parameter, while GA-DAN [65] models cross-

domain shifts in geometry and appearance. MC-GAN [2] facilitates font style transfer for letters A to Z, and Wu et al. [58] developed an end-to-end trainable style retention network for editing text in natural photos. Although diffusion models [14, 47, 68] have shown impressive generation capabilities, they struggle with precise text rendering. TextDiffuser [9] addresses this by using a customized text dataset with OCR annotations, creating visually appealing text coherent with backgrounds. However, this approach relies on expensive OCR tools and paired image-text data, making it less ideal for real-world text editing.

Text Style Transfer: The task of transferring an image's visual style from a reference image to a target image is known as image style transfer. Many existing techniques use an encoder-decoder architecture to embed and then decode the input into a desired output. Isola et al. [26] developed a learnable mapping using aligned image pairings, while Zhu et al. [72] introduced cycle consistency loss to generalize mappings for unpaired data. Challenges like creating images from sketches and synthesizing faces have been approached similarly. An algorithm proposed by Yang et al. [61] uses statistical measurements to transfer text effects by modeling distance-based properties. Additionally, Yang et al. [62] utilize stylization and de-stylization sub-networks for style transfer and removal. Other works integrate background and text style information for artifact-free text editing [52], focus on text image generation with typographic attributes [25, 53], and explore style and content disentanglement for OCR-based recognition [30], along with attribute-conditioned text-style transfer [17, 34, 42, 59]. This work aims to enhance model efficiency in style transfer.

Scene Text Editing: The variety of uses for GAN-based scene text editing has piqued researchers' growing curiosity. To alter a single character, for instance, Roy et al. [49] created the Font Adaptive Neural Network (STEFANN). This character-level alteration, however, falls short of replacing words with length alterations, which restricts practical uses. In [58], the authors developed the word-level editing approach employing three sub-networks: background inpainting, text conversion, and fusion to overcome this constraint. Each module can manage a reasonably straightforward task thanks to the divide-and-conquer method. To make the text conversion module easier to learn, [60] enhanced SRNet and added the TPS module, which separates the spatial transformation from text styles. Furthermore, [71] suggested the forge-and-recapture procedure to reduce visual artifacts and applied scene text editing to document forging. To keep the consistency of all other edited frames, [55] used SRNet for video text editing, altering only the chosen frame as a reference and applying certain photometric modifications. These methods, which are essentially extensions of SRNet, can only be trained on artificial datasets and may not be able to replace text with complicated styles. A stroke-level alteration technique that creates more readable text graphics are suggested by [44]. Their technique can be trained on both labeled synthetic datasets and unpaired scene text images, and it supports the semi-supervised training scheme. Recently, stable diffusion-based approaches [8, 28, 39] have been investigated for text editing in natural scenes and handwriting domain. This work conducts a rigorous comparison with [28] on the STE task.

3. Proposed Methodology

In this section, we will examine the proposed methodology of FASTER, including the problem formulation, the two stages (task modules) used in the framework, and the optimal learning objectives utilized in the architecture.

3.1. Problem Formulation

Given a scene text image I_A, the objective of the proposed STE is to generate an image I_B with a modified text. To enforce a classifier-free image translation, we aim to condition the generative process on the structural guidance (m_A, m_B), where m_A and m_B correspond to the binary masks of text content in I_A and I_B, respectively. However, obtaining m_B before generating I_B is unrealistic, making the end-to-end text style transfer difficult. We address this issue by splitting the generative process into two independent stages containing the *Style Translation Block (STB)* and *Content Translation Block (CTB)*, respectively.

- In *Stage-I*, we replace the initially unknown mask m_B with another mask m_F having the same textual content but in a fixed font of known style. In this stage, the generator G_m produces an approximation of the target style mask $\overline{m_B}$.
- In *Stage-II*, an identical generator G_i synthesizes the approximate target image $\overline{I_B}$ by transferring attributes from I_A and using $(m_A, \overline{m_B})$ as structural guidance. We use synthetically generated (I_A, m_A) pairs to train both G_m and G_i.
- Additionally, a U-Net [48] *feature extraction backbone* inspired from [50] has been separately trained for estimating the mask $\overline{m_A}$ from I_A during inference on real scene text samples. The entire overview of our architecture is shown in Figure 2.

3.2. Stage – I: Target Style Mask Estimation

The Stage-I architecture includes a functional STB module which is a GAN network containing a target mask generator G_m and a PatchGAN [26] discriminator D_m. G_m takes the source image I_A and the channel-wise concatenated masks $m_\theta = (m_A, m_F)$ as inputs and produces an approximate target style mask $\overline{m_B}$ as the output. The discriminator D_m discriminates between real and fake transformations by processing channel-wise concatenated masks (m_A, m_B) or $(m_A, \overline{m_B})$, predicting a binary class probability map of ones (real) or zeros (fake).

Encoder: G_m comprises two encoding branches for I_A and

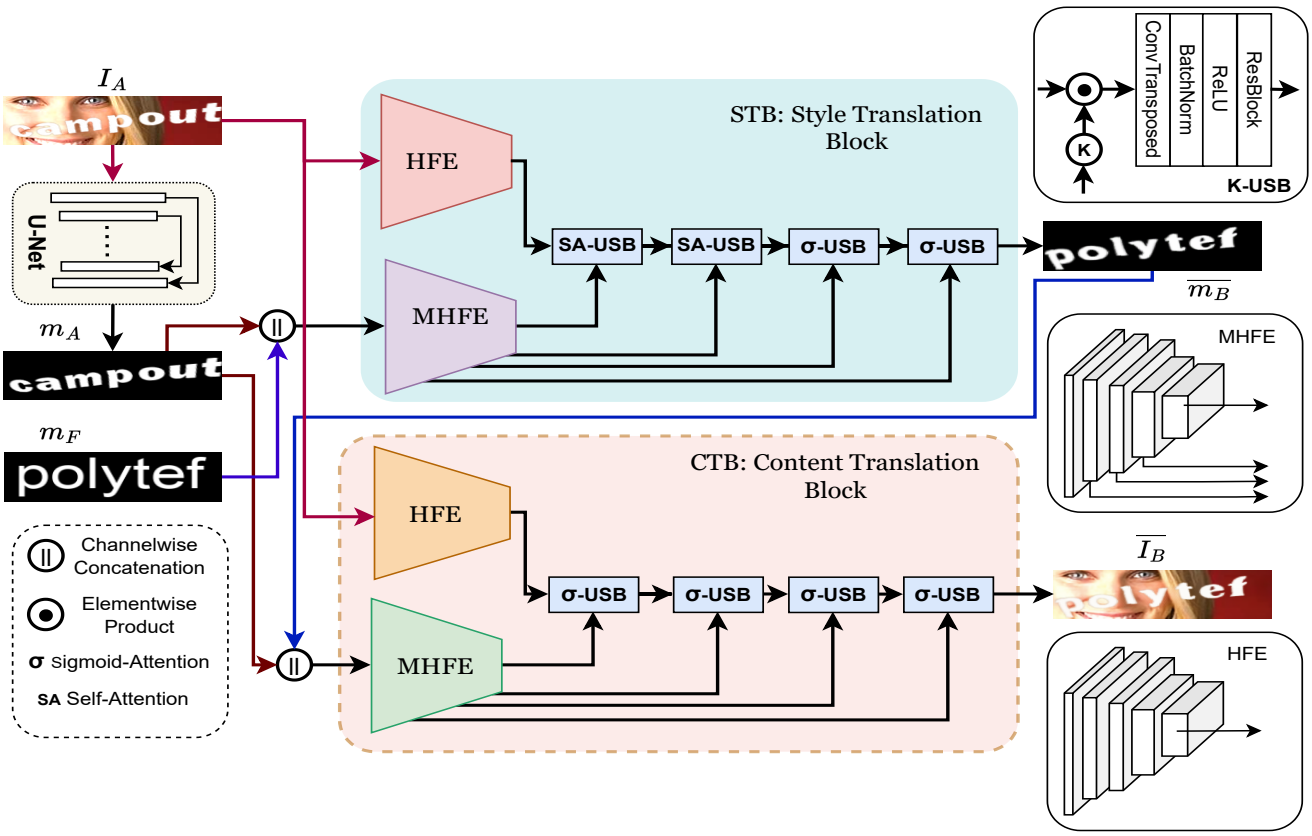

Figure 2. **Overall Architecture of FASTER**. In **Stage-I**, an approximate target style mask $\overline{m_B}$ is estimated from the source image I_A, source style mask m_A, and a fixed style mask m_F of the target text. In **Stage-II**, the target image $\overline{I_B}$ is generated by transferring image attributes from I_A and conditioning the image translation on the structural guidance $(m_A, \overline{m_B})$.

m_θ to be referred as *Hierarchical Feature Extractor (HFE)* and *Multi-scale Hierarchical Feature Extractor (MHFE)*. The condition image and the guidance masks are resized to a dimension of 64×256. At every branch (HFE and MHFE), the encoder first projects the input into a 64-channel feature space using 3×3 convolution kernels with stride 1, padding 1, and without adding any bias. The final block output is obtained after batch normalization (BN) and ReLU units. The projected feature space is then downscaled four times. At every downsampling block, the spatial feature dimension is downscaled to half while doubling the number of channels. Each downsampling block uses 4×4 convolution kernels, stride 1, padding 1, and zero bias. Each downscaling convolution is immediately followed by BN, ReLU and a basic residual block [20] to generate the final block output.

Decoder: During decoding, the outputs of both encoding branches HFE and MHFE are channel-wise concatenated and passed through the decoder. The decoder consists of four *UpScaling Blocks (USB)*, each doubling the spatial feature dimension while decreasing the number of channels to half. The upscaling is performed with 4×4 transposed convolution kernels, stride 1, padding 1, and zero bias. Like the encoder, the final block output is generated following BN, ReLU activation, and a basic residual block. In G_m, we apply two different attention mechanisms at matching feature resolutions of the encoder and decoder to attend to both coarse and fine image attributes during structural transformation. At the two lowest resolutions, we use *cascaded self-attention* similar to [67], and at the other two higher resolutions, we use a *sigmoid attention*. The self-attention units at the two lowest resolutions $k = \{1, 2\}$ can represented mathematically as:

$$m_1^{\phi_m} = \phi_{m1}(I_4^{\varphi_m} \odot SA(m_4^{\varphi_m}))$$

$$m_2^{\phi_m} = \phi_{m2}(m_1^{\phi_m} \odot SA(m_3^{\varphi_m}))$$

and for the following decoder blocks at higher resolutions $k = \{3, 4\}$ that use sigmoid attention can be represented as:

$$m_k^{\phi_m} = \phi_{mk}(m_{k-1}^{\phi_m} \odot \sigma(m_{5-k}^{\varphi_m}))$$

where, $I_k^{\varphi_m}$ is the output of k-th image encoder block, $m_k^{\varphi_m}$ is the output of k-th mask encoder block, $m_k^{\phi_m}$ is the output of the k-th decoder block, SA and σ denote *self-attention* and *sigmoid attention*, respectively, and $\odot$ denotes element-wise product. The output feature maps from the decoder are post-processed through four consecutive basic

residual blocks. The resulting feature space is projected to a 3-channel image space of spatial resolution of 64×256 by a point convolution with 1×1 kernel, unit stride, zero padding, and without bias. The final normalized output of G_m is obtained following a hyperbolic tangent ($\mathbf{tanh}$) activation function.

3.3. Stage – II: Text Style Transfer with Structural Guidance

The Stage II architecture is identical to Stage-I, with slightly different input specifications and attention mechanisms. In this case, the generator G_i takes the source image I_A and the channel-wise concatenated masks $(m_A, \overline{m_B})$ as inputs and produces an approximate target image $\overline{I_B}$ as the output. The PatchGAN discriminator D_i discerns between a real and a fake transformation by taking channel-wise concatenated images (I_A, I_B) or $(I_A, \overline{I_B})$ and predicting a binary class probability map of ones (real) or zeroes (fake).

In G_i, we apply only *sigmoid attention* at every matching feature resolution of the encoder and decoder. Mathematically, at the lowest resolution $k = 1$,

$$I_1^{\phi_i} = \phi_{i1}(I_4^{\varphi_i} \odot \sigma(m_4^{\varphi_i}))$$

and for the following decoder blocks at higher resolutions $k = \{2, 3, 4\}$,

$$I_k^{\phi_i} = \phi_{ik}(I_{k-1}^{\phi_i} \odot \sigma(m_{5-k}^{\varphi_i}))$$

where, $I_k^{\varphi_i}$ is the output of k-th image encoder block, $m_k^{\varphi_i}$ is the output of k-th mask encoder block, $I_k^{\phi_i}$ is the output of the k-th decoder block, σ denotes *sigmoid attention*, and $\odot$ denotes element-wise product.

3.4. Learning Objectives

Stage – I Objectives: The optimization objective of generator G_m consists of four different loss components – (a) pixel-wise L_2 loss $\mathcal{L}_{L_2}^{G_m}$, (b) discriminator loss $\mathcal{L}_{GAN}^{G_m}$ estimated by the discriminator D_m, (c) perceptual loss $\mathcal{L}_{P_\rho}^{G_m}$ computed using a pre-trained VGG-19 network [54], (d) multi-scale structural similarity [57] loss $\mathcal{L}_{SSIM}^{G_m}$, and (e) OCR perceptual loss [46] $\mathcal{L}_{OCR_{per}}^{G_m}$ computed using pre-trained CRAFT [4] model. The L_2 reconstruction loss is estimated as the mean squared error (MSE) between the generated style mask $\overline{m_B}$ and target style mask m_B. The GAN discriminator objective is defined as the binary cross-entropy (BCE) estimated by D_m computed between predicted mask $\overline{m_B}$ and the input style mask m_A. The image perceptual loss $\mathcal{L}_{P_\rho}^{G_m}$ computed between target style mask m_B and generated style mask $\overline{m_B}$ follows a similar pattern as [29]. We include two perceptual loss terms for $\rho = 4$ and $\rho = 9$ in the final objective function. - The multi-scale structural similarity loss $\mathcal{L}_{SSIM}^{G_m}$ is estimated similarly as [57] between target style mask m_B and generated style mask $\overline{m_B}$.

- The OCR perceptual loss follows a similar pattern as in [46] between target style mask m_B and generated style mask $\overline{m_B}$ is defined as:

$$\mathcal{L}_{OCR_{per}}^{G_m} = \sum_{a=1}^{\rho} \frac{1}{h_\rho w_\rho} \sum_{h,w} \left\| (\overline{m_B}_{h,w}^{\rho}) - {m_B}_{h,w}^{\rho} \right\|_2^2 \quad (1)$$

where $\overline{m_B}_{h,w}^{\rho}$ and ${m_B}_{h,w}^{\rho}$ are the activation map of layer ρ has been taken from the pretrained CRAFT [4] model, and $\| \cdot \|_2^2$ denotes the L_2 norm (mean squared error).
- The overall objective function G_m of the STB functional module can be given by:

$$\begin{aligned} \mathcal{L}_{G_m} = \lambda_1 \cdot \mathcal{L}_{L_2}^{G_m} + \lambda_2 \cdot \mathcal{L}_{GAN}^{G_m} + \\ \lambda_3 \cdot (\mathcal{L}_{P_4}^{G_m} + \mathcal{L}_{P_9}^{G_m}) + \lambda_4 \cdot \mathcal{L}_{SSIM}^{G_m} + \lambda_5 \cdot \mathcal{L}_{OCR_{per}}^{G_m} \end{aligned} \quad (2)$$

where λ_1, λ_2, λ_3, λ_4, and λ_5 are the weighing parameters for respective loss components.
- The optimization objective of the STB discriminator D_m is given by:

$$\begin{aligned} \mathcal{L}_{D_m} = \frac{1}{2}[\mathcal{L}_{BCE}(D_m(m_A, m_B), 1) + \\ \mathcal{L}_{BCE}(D_m(m_A, \overline{m_B}), 0)] \end{aligned} \quad (3)$$

Stage – II Objectives: Similarly, the final optimization objective of generator G_i for the CTB functional module consists of four loss components – (a) pixel-wise L_1 loss $\mathcal{L}_{L_1}^{G_i}$ between target image I_B and generated image $\overline{I_B}$, (b) discriminator loss $\mathcal{L}_{GAN}^{G_i}$ estimated by the discriminator D_i between target image I_B and generated image $\overline{I_B}$, (c) perceptual loss $\mathcal{L}_{P_\rho}^{G_i}$ computed using a pre-trained VGG-19 network between target image I_B and generated image $\overline{I_B}$, and (d) OCR perceptual loss $\mathcal{L}_{OCR_{per}}^{G_i}$ computed using the pretrained CRAFT model between target image I_B and generated image $\overline{I_B}$.
- The overall objective function of the CTB GAN generator G_i is given by:

$$\begin{aligned} \mathcal{L}_{G_i} = \beta_1 \cdot \mathcal{L}_{L_2}^{G_i} + \beta_2 \cdot \mathcal{L}_{GAN}^{G_i} + \\ \beta_3 \cdot (\mathcal{L}_{P_4}^{G_i} + \mathcal{L}_{P_9}^{G_i}) + \beta_4 \cdot (\mathcal{L}_{OCR_{per}}^{G_i}) \end{aligned} \quad (4)$$

where β_1, β_2, β_3, and β_4 are the weighing parameters for respective loss components. The optimization objective of the CTB discriminator D_i is given by:

$$\begin{aligned} \mathcal{L}_{D_i} = \frac{1}{2}[\mathcal{L}_{BCE}(D_i(I_A, I_B), 1) + \\ \mathcal{L}_{BCE}(D_i(I_A, \overline{I_B}), 0)] \end{aligned} \quad (5)$$

We have used $\lambda_1 = 3$, $\lambda_2 = 1$, $\lambda_3 = 3$, $\lambda_4 = 3$, $\lambda_5 = 10$, $\beta_1 = 5$, $\beta_2 = 1$, $\beta_3 = 5$ and $\beta_4 = 5$ as the weighing parameters for our experimental setup. The values of these weighing parameters have been empirically estimated through an extensive ablation study.

Table 1. Quantitative comparison with SOTA. Results style: **best**, second best. ↑ higher is better and ↓ lower is better. Method[1] has been pretrained on the same data.

Method	pix2pix [26]	SRNet[1] [58]	SwapText [60]	MOSTEL[1] [44]	DiffSTE [28]	**FASTER**[1]	Δ
PSNR ↑	12.01	18.66	19.43	20.75	13.40	**20.84**	**+0.09**
SSIM ↑	0.349	0.614	0.652	0.707	0.3886	**0.790**	**+0.083**
FID ↓	164.24	48.16	35.62	37.55	65.06	**11.79**	**-23.83**
$LPIPS_1$ ↓	-	0.2076	-	0.1770	0.3958	**0.0955**	**-0.0815**
$LPIPS_2$ ↓	-	0.3779	-	0.2895	0.5341	**0.1782**	**-0.1113**
$LPIPS_3$ ↓	-	0.2524	-	0.2275	0.4477	**0.1227**	**-0.1048**
Acc (OCR) ↑	-	14.79	-	20.10	11.90	**30.30**	**+10.2**
NED (OCR) ↑	-	0.414	-	0.563	**0.714**	0.660	**-0.054**
Clip Score ↑	-	22.55	-	23.74	**25.56**	24.00	**-1.56**
Inf. Time ↓	-	46.88	-	76.87	12000	**31.12**	**-15.76**

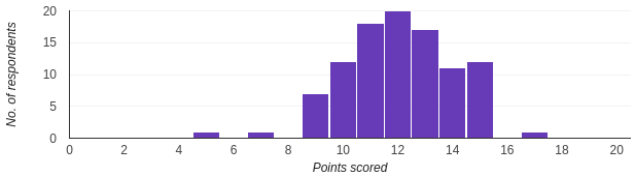

Figure 3. Human Evaluation Study

4. Experiments

4.1. Datasets

Synthetic Data: For the supervised training of our pipeline, we utilized a dataset of 150,000 labeled images as mentioned in the work of Qu et al. [44]. For evaluation, we use the Tamper-Syn2k [44] dataset, which consists of 2,000 paired images specifically designed for evaluation purposes. We also generated an additional set of 100,000 labeled images using 2,500 fonts. This expanded dataset was employed specifically for training purposes of the proposed method.
Real Data: In our research on enhancing natural scene text editing on real scene images, we utilized a dataset consisting of real scene images obtained from the MOSTEL paper [44]. This dataset is generated by the authors using random cropped images from ICDAR 2013 [32], MLT-2017 [38] MLT-2019 [37] datasets.

4.2. Implementation Details

The pre-transformation stage includes resizing the input images to 64×256. We utilize the Adam optimizer [33] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and set the learning rate to 1×10^{-3} for both the Stages. We train FASTER for a total of 100k iterations using a batch size of 40. Our pipeline is implemented using PyTorch [24] and it is trained on a single NVIDIA 4080 OC GPU.

4.3. Evaluation Metrics

To evaluate the effectiveness of FASTER in the STE task, we employ the following commonly used metrics adapted in [58, 60] to evaluate the generated target style edited image: (1) `Mean Squared Error (MSE)`, which measures the L-2 distance between the image pair; (2) `Peak Signal-to-Noise Ratio` (PSNR), which measures the text image quality, and (3) `Structural Similarity Index Measure` (SSIM) [57], which measures the mean structural similarity between the target and edited sample. We have also used the GAN metrics adopted in [34] for further evaluation for (4) `Learned Perceptual Image Patch Similarity` (LPIPS) [69], which measures the similarities in activation of the paired image patches of Squeezenet [23], VGG [54] and AlexNet [35] denoted as $LPIPS_1$, $LPIPS_2$ and $LPIPS_3$ in Table 1. (5) `Frechet Inception Distance (FID)` computes the distance between the feature vectors of the target and edited text images.(6) `OCR Accuracy and Normalized Edit Distance` [3], which calculates the recognition accuracies from the image patches. (7) `Clip Score` Introduced in [21], it computes the semantic similarity between the image and the textual space as in CLIP model [45] and quantifies "compatibility". The reason CLIP Score has been adapted for evaluation is due to its high correlation with human judgment and inspired from [9].

4.4. Human Evaluation Study

Human evaluation studies provide crucial insights into how edited and rendered text images are perceived and utilized in real-world scenarios, ensuring that STE models meet end-user expectations. An interesting human evaluation study was conducted with 100 human participants independently asked to mark some samples of images as real (original) or fake (edited). We computed how many images generated by FASTER have been identified as real by the participants. Figure 3 shows a plot of the points scored by participants based on correctly identified images. An **overall human misclassification rate of 39.75%** was recorded which shows the difficulty of the fake identification challenge for the participants created by FASTER. Figure 11 shows some of

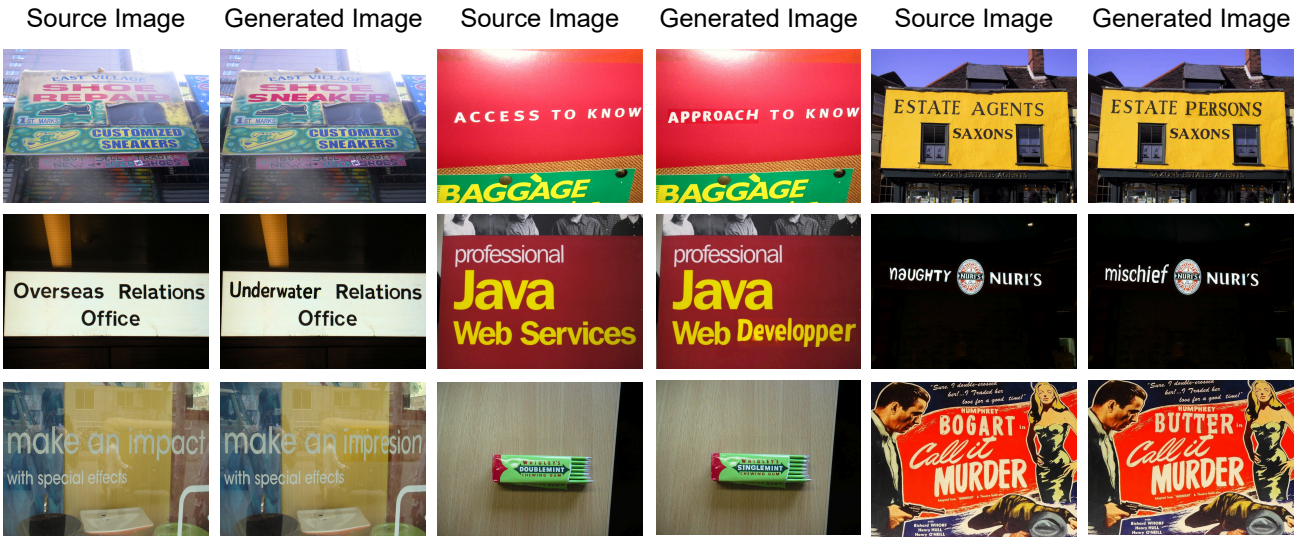

Figure 4. Visual STE results with FASTER on samples from the Real dataset, selected from the human evaluation study with the highest number of incorrect predictions. Please zoom 300% for better visualization.

the qualitative examples assigned to the user study where participants failed mostly to classify correctly.

4.5. Comparisons with Existing SOTA

We present a detailed quantitative and qualitative comparison of our proposed method, FASTER, with the existing SOTA methods in STE [28, 44, 58, 60] and the classic pix2pix [26] approach. Our model has been trained on the same dataset as used in the MOSTEL paper [44] and our own synthetically generated data.

Quantitative Analysis: As shown in Table 1, results demonstrate that our proposed method (FASTER) outperforms all the SOTA approaches in 8 out of the 10 evaluation measures. The key highlights and insights from the results are:

- There has been a *substantial improvement in the perceptual FID Score* with a difference of 23.83 over second-best SwapText. This indicates that our approach generates images with higher similarity to the ground truth and exhibits better perceptual quality.
- The OCR *recognition accuracy* scores have also had a *10% gain* over second-best MOSTEL, which shows that generated images have excellent legibility and that generated images blend seamlessly with edited content.
- We also evaluated with the *inference time* and compared our method with other approaches. FASTER performs editing operations with 15.76 milliseconds less than the second-best SRNet. This justifies the efficiency of the model, especially in sharp contrast with stable diffusion approaches like DiffSTE which takes 12000 milliseconds to operate.
- The DiffSTE approach shows the best results in terms of ClipScore and Normalized ED (OCR) metrics as they have adapted the Clip model [45] in their training pipeline, which aligns the semantic space between the image and text.

Qualitative Analysis: As shown in Figure 6, we provide visual comparisons of the generated images from different previous methods and FASTER. Notably, the images generated by SRNet demonstrate difficulties in accurately capturing the desired features. While MOSTEL's generated images are of good quality, when compared directly with FASTER, they are not as impressive. It is important to note that although MOSTEL shows better results in some cases compared to FASTER, the visual quality and fidelity of the generated images produced by FASTER, surpass MOSTEL's performance. The most interesting results have been obtained with DiffSTE approaches where the text does not blend well with the image background, which shows the incapability of Stable diffusion models when it comes to rendering precise (fine-grained) image objects like text in scenes. On the contrary, as shown in Figure 11 the editing results in real data for the FASTER model show that it is highly robust with background distortions. Overall, FASTER showcases superior results both quantitatively and qualitatively when compared to the previous methods considered in the evaluation.

4.6. Ablation Study

An exhaustive set of experiments has been conducted with the FASTER framework to investigate how the integrated model components have contributed to the overall potential.

Effectiveness of Individual Learning Objectives: To show the individual importance of the learning objectives and how they impact final STE performance, we conducted experiments as shown in Table 2. It throws some quantitative insights in terms of different metrics. Adding OCR Perceptual loss [46] term to the training objective helps minimise the distance between the target and edited text samples in the feature space. Table 2 show that $\mathcal{L}_{OCR_{per}}$ *has the high-*

Figure 5. **Left** Fancy Font Text Editing. **Right** Visual comparison with TextDiffuser and DiffSTE on larger context

Table 2. Effectiveness of individual losses on FASTER. Results style: **best**, second best. Results style: **best**, second best.

Method	MSE ↓	PSNR ↑	SSIM ↑	FID ↓	$LPIPS_1$ ↓	$LPIPS_2$ ↓	$LPIPS_3$ ↓	Acc ↑	NED ↑	Clip Score ↑
w/o $\mathcal{L}_{GAN}$	0.0200	18.41	0.6889	23.56	0.1332	0.2431	0.1651	30.65	0.654	23.89
w/o $\mathcal{L}_{l2}^{G_m}$	0.0200	18.40	0.6880	19.58	0.1327	0.2436	0.1657	29.40	0.652	23.90
w/o $\mathcal{L}_{SSIM}^{G_m}$	0.0198	18.47	0.6907	23.47	0.1309	0.2409	0.1633	30.75	0.655	23.90
w/o $\mathcal{L}_{P_r}$	0.0201	18.40	0.6878	23.60	0.1327	0.2441	0.1653	**31.15**	**0.662**	23.78
w/o $\mathcal{L}_{OCR_{per}}$	0.0224	18.00	0.6775	24.59	0.1441	0.2598	0.1782	19.20	0.569	23.89
All	**0.0127**	**20.85**	**0.7905**	**11.79**	**0.0955**	**0.1782**	**0.1227**	30.30	0.660	**24.00**

Table 3. Effectiveness of Data Mixing.

Dataset	MSE ↓	PSNR ↑	SSIM ↑
MOSTEL [44]	0.0171	19.04	0.707
Mixed	**0.0162**	**19.19**	**0.718**

Table 4. Effect of different USBs.

Attentions in USB						
1st	2nd	3rd	4th	MSE ↓	PSNR ↑	SSIM ↑
σ	σ	σ	σ	0.0216	18.22	0.690
SA	SA	SA	SA	0.0213	18.00	0.672
SA	SA	σ	σ	**0.0171**	**19.04**	**0.707**

Table 5. Effectiveness of Inputs.

Input Combination				
Input Type	m_a Concat	MSE ↓	PSNR ↑	SSIM ↑
Mask	Concat	0.0162	19.19	0.718
Mask	w/o Concat	0.0178	18.86	0.701
Image	Concat	**0.0135**	**20.20**	**0.776**
Image	w/o Concat	0.0156	19.55	0.759

Table 6. Effectiveness of different Fonts.

Fonts	MSE ↓	PSNR ↑	SSIM ↑
Sans Serif	0.0270	17.16	0.6432
Times New Roman	0.0273	17.11	0.6372
Arial	**0.0264**	**17.27**	**0.6493**

Table 7. Effectiveness of font sizes.

Method	MSE ↓	PSNR ↑	SSIM ↑
Arial font size 28	0.0283	16.90	0.6353
Arial font size 25	**0.0264**	**17.27**	**0.6493**

Table 8. Effectiveness of two stages.

Method	MSE ↓	PSNR ↑	SSIM ↑
Jointly	0.0311	17.14	0.6773
Independently	**0.0135**	**20.20**	**0.7760**

est impact on overall performance metrics (especially FID, PSNR and MSE scores). While the VGG perceptual loss also leaves a reasonably good individual impact, it is interesting to note that it decreases the content evaluation metrics (OCR Accuracy and NED). This can be attributed to the fact that it adds noise to the already high-performant OCR perceptual loss.

Effectiveness of Data Mixing: In this study, we utilized two datasets: the MOSTEL dataset for initial training and a mixed dataset combining MOSTEL and SRNET data. Results from Table 3 demonstrate that the mixed data model performs better than the single dataset model.

How effective is the Cascaded Self-Attention in USBs? We investigated the influence of different attention mechanisms on image upsampling in FASTER. Initially, we evaluated the performance of isolated sigmoid attention by incorporating four sigmoid attention blocks in the USBs. Subsequently, we examined the integration of four self-attention blocks within the USB. Finally, combining two sigmoid and two self-attention blocks yielded superior metrics compared to previous experiments, as shown in Table 4 using the dataset in [44].

Concatenation Strategies and Input Combination: Initially, our proposed pipeline was guided by concatenating the input mask m_A with a fixed mask m_F. Then, we experimented to assess the outcome when only the fixed mask m_F was utilized without concatenation. Additionally, we evaluated the effectiveness of producing results from both binary source mask and image I_A as shown in Table 5.

Exploring Font Style and Size Variations: As shown in Table 6 and Table 7, we conducted an ablation study to investigate the impact of different font types and sizes on FASTER's performance. The results indicate that varying fonts (Arial, Times New Roman, Sans Serif) and sizes (25 and 28) have minimal impact. Our synthetic training dataset includes diverse fonts and sizes, similar to a standard OCR dataset. These findings highlight FASTER's robustness in handling diverse font styles and sizes.

FASTER benefits more from independent stage training compared to joint training. FASTER consist of two stages (STB and CTB) in the overall pipeline. As demonstrated in Table 8, The "Two stages Independently" strategy showcases noteworthy outcomes with an impressive MSE reduction to 0.013, a notable PSNR increase to 20.20, and a significant SSIM enhancement to 0.77. By training style and content blocks independently, each block can specialize in its respective task, leading to more effective learning and adaptation to the characteristics of the input data compared to less favourable "Two stages jointly".

5. Conclusion and Future Scope

Current STE methods often prioritize image quality over practical usability. To address this, we introduce FASTER, a two-stage GAN-based framework that seamlessly edits and

renders text while preserving its natural appearance. The cascaded self-attention module enhances processing speed and editing capabilities. FASTER is robust across various text attributes (font type, size, alignment) and domains (scene text, documents, handwriting), making it ideal for real-world applications.

The main limitation of our method is its reliance on mask guidance maps to identify editing regions. While this helps eliminate background distractions and focus on text regions, it struggles with complex or irregular text layouts. The guidance maps assume well-defined masks, which is problematic for diverse text forms like complex handwriting, irregular, or overlapping text (see Supplementary). Generating accurate masks in these cases can be challenging and may lead to errors.

References

[1] Gantugs Atarsaikhan, Brian Kenji Iwana, Atsushi Narusawa, Keiji Yanai, and Seiichi Uchida. Neural font style transfer. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 5, pages 51–56. IEEE, 2017. 2

[2] Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7564–7573, 2018. 2, 3

[3] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4715–4723, 2019. 6

[4] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9365–9374, 2019. 5

[5] Sanket Biswas, Rajiv Jain, Vlad I. Morariu, Jiuxiang Gu, Puneet Mathur, Curtis Wigington, Tong Sun, and Josep Lladós. Docsynthv2: A practical autoregressive modeling for document generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 8148–8153, June 2024. 2

[6] Kim Sydow Campbell. *Coherence, continuity, and cohesion: Theoretical foundations for document design*. Routledge, 2013. 2

[7] Jacky Cao, Kit-Yung Lam, Lik-Hang Lee, Xiaoli Liu, Pan Hui, and Xiang Su. Mobile augmented reality: User interfaces, frameworks, and intelligence. *ACM Computing Surveys*, 55(9):1–36, 2023. 2

[8] Haoxing Chen, Zhuoer Xu, Zhangxuan Gu, Yaohui Li, Changhua Meng, Huijia Zhu, Weiqiang Wang, et al. Diffute: Universal text editing diffusion model. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[9] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *arXiv preprint arXiv:2305.10855*, 2023. 2, 3, 6

[10] Alloy Das, Sanket Biswas, Ayan Banerjee, Josep Lladós, Umapada Pal, and Saumik Bhattacharya. Harnessing the power of multi-lingual datasets for pre-training: Towards enhancing text spotting performance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 718–728, 2024. 2

[11] Alloy Das, Sanket Biswas, Umapada Pal, and Josep Lladós. Diving into the depths of spotting text in multi-domain noisy scenes. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 410–417. IEEE, 2024. 2

[12] Alloy Das, Sanket Biswas, Umapada Pal, Josep Lladós, and Saumik Bhattacharya. Fasttextspotter: A high-efficiency transformer for multilingual scene text spotting. *arXiv preprint arXiv:2408.14998*, 2024. 2

[13] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Acm siggraph 2008 classes*, pages 1–10, 2008. 2

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3

[15] Jun Du, Qiang Huo, Lei Sun, and Jian Sun. Snap and translate using windows phone. In *2011 International Conference on Document Analysis and Recognition*, pages 809–813. IEEE, 2011. 2

[16] Victor Fragoso, Steffen Gauglitz, Shane Zamora, Jim Kleban, and Matthew Turk. Translatar: A mobile augmented reality translator. In *2011 IEEE workshop on applications of computer vision (WACV)*, pages 497–502. IEEE, 2011. 2

[17] Raul Gomez, Ali Furkan Biten, Lluis Gomez, Jaume Gibert, Dimosthenis Karatzas, and Marçal Rusiñol. Selective style transfer for text. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 805–812. IEEE, 2019. 3

[18] I Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *(Advances in neural information processing systems)(pp. 2672–2680)*, 2014. 2

[19] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[21] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 6

[22] Qirui Huang, Bin Fu, Yu Qiao, et al. Gentext: Unsupervised artistic text generation via decoupled font and texture manipulation. *arXiv preprint arXiv:2207.09649*, 2022. 2

[23] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 6

[24] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104, 2021. 6, 12

[25] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Towards flexible multi-modal document models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14287–14296, 2023. 3

[26] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 3, 6, 7

[27] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. 2

[28] Jiabao Ji, Guanhua Zhang, Zhaowen Wang, Bairu Hou, Zhifei Zhang, Brian Price, and Shiyu Chang. Improving diffusion models for scene text editing with dual encoders. *arXiv preprint arXiv:2304.05568*, 2023. 2, 3, 6, 7, 13

[29] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 5

[30] Lei Kang, Pau Riba, Marcal Rusinol, Alicia Fornes, and Mauricio Villegas. Content and style aware generation of text-line images for handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8846–8860, 2021. 3

[31] Levent Karacan, Zeynep Akata, Aykut Erdem, and Erkut Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215*, 2016. 2

[32] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th international conference on document analysis and recognition*, pages 1484–1493. IEEE, 2013. 6, 12

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6, 12

[34] Praveen Krishnan, Rama Kovvuri, Guan Pang, Boris Vassilev, and Tal Hassner. Textstylebrush: Transfer of text aesthetics from a single example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2, 3, 6

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 6

[36] Junyeop Lee, Yoonsik Kim, Seonghyeon Kim, Moonbin Yim, Seung Shin, Gayoung Lee, and Sungrae Park. Rewritenet: Reliable scene text editing with implicit decomposition of text contents and styles. *arXiv preprint arXiv:2107.11041*, 2021. 2

[37] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *2019 International conference on document analysis and recognition (ICDAR)*, pages 1582–1587. IEEE, 2019. 6, 12

[38] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 1454–1459. IEEE, 2017. 6, 12

[39] Konstantina Nikolaidou, George Retsinas, Giorgos Sfikas, and Marcus Liwicki. Diffusionpen: Towards controlling the style of handwritten text generation. In *European Conference on Computer Vision*, 2024. 3

[40] Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang. On the integration of self-attention and convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–825, 2022. 2

[41] Song Park, Sanghyuk Chun, Junbum Cha, Bado Lee, and Hyunjung Shim. Multiple heads are better than one: Few-shot font generation with multiple localized experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13900–13909, 2021. 2

[42] Vittorio Pippi, Silvia Cascianelli, and Rita Cucchiara. Handwritten text generation from visual archetypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22458–22467, June 2023. 3

[43] Xugong Qin, Yu Zhou, Youhui Guo, Dayan Wu, Zhihong Tian, Ning Jiang, Hongbin Wang, and Weiping Wang. Mask is all you need: Rethinking mask r-cnn for dense and arbitrary-shaped scene text detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 414–423, 2021. 2

[44] Yadong Qu, Qingfeng Tan, Hongtao Xie, Jianjun Xu, Yuxin Wang, and Yongdong Zhang. Exploring stroke-level modifications for scene text editing. *arXiv preprint arXiv:2212.01982*, 2022. 2, 3, 6, 7, 8, 12, 13

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6, 7

[46] Juan A Rodriguez, David Vazquez, Issam Laradji, Marco Pedersoli, and Pau Rodriguez. Ocr-vqgan: Taming text-within-image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3689–3698, 2023. 5, 7

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2, 3

[49] Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, and Umapada Pal. Stefann: scene text editor using font adaptive neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13228–13237, 2020. 2, 3

[50] Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, and Umapada Pal. Multi-scale attention guided pose transfer. *Pattern Recognition*, 137:109315, 2023. 3

[51] Yangming Shi, Haisong Ding, Kai Chen, and Qiang Huo. Aprnet: Attention-based pixel-wise rendering network for photo-realistic text image generation. *arXiv preprint arXiv:2203.07705*, 2022. 2

[52] Wataru Shimoda, Daichi Haraguchi, Seiichi Uchida, and Kota Yamaguchi. De-rendering stylized texts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1076–1085, 2021. 3

[53] Wataru Shimoda, Daichi Haraguchi, Seiichi Uchida, and Kota Yamaguchi. Towards diverse and consistent typography generation. *arXiv preprint arXiv:2309.02099*, 2023. 3

[54] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. 3rd int conf learn represent iclr 2015-conf track proc. september 2014: 1–14, 2014. 5, 6

[55] Jeyasri Subramanian, Varnith Chordia, Eugene Bart, Shaobo Fang, Kelly Guan, Raja Bala, et al. Strive: Scene text replacement in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14549–14558, 2021. 3

[56] Qi Wan, Haoqin Ji, and Linlin Shen. Self-attention based text knowledge mining for text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5983–5992, June 2021. 2

[57] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5, 6

[58] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Editing text in the wild. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1500–1508, 2019. 2, 3, 6, 7, 13

[59] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5481–5489, 2021. 3

[60] Qiangpeng Yang, Jun Huang, and Wei Lin. Swaptext: Image based texts transfer in scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14700–14709, 2020. 2, 3, 6, 7

[61] Shuai Yang, Jiaying Liu, Zhouhui Lian, and Zongming Guo. Awesome typography: Statistics-based text effects transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7464–7473, 2017. 3

[62] Shuai Yang, Jiaying Liu, Wenjing Wang, and Zongming Guo. Tet-gan: Text effects transfer via stylization and destylization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1238–1245, 2019. 3

[63] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable artistic text style transfer via shape-matching gan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4442–4451, 2019. 2

[64] Fangneng Zhan, Shijian Lu, and Chuhui Xue. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 249–266, 2018. 2

[65] Fangneng Zhan, Chuhui Xue, and Shijian Lu. Ga-dan: Geometry-aware domain adaptation network for scene text detection and recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9105–9115, 2019. 2

[66] Fangneng Zhan, Hongyuan Zhu, and Shijian Lu. Spatial fusion gan for image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3653–3662, 2019. 2

[67] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019. 2, 4

[68] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 3

[69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[70] Yexun Zhang, Ya Zhang, and Wenbin Cai. Separating style and content for generalized style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8447–8455, 2018. 2

[71] Lin Zhao, Changsheng Chen, and Jiwu Huang. Deep learning-based forgery attack on document images. *IEEE Transactions on Image Processing*, 30:7964–7979, 2021. 3

[72] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 3

Supplementary Material for FASTER

Synthetic Data: For the supervised training of our pipeline, we utilized a dataset of 150,000 labeled images as mentioned in the work of Qu et al. [44]. For evaluation, we use the Tamper-Syn2k [44] dataset, which consists of 2,000 paired images specifically designed for evaluation purposes. These paired images were created by rendering different texts with consistent styles, including font, size, color, spatial transformation, and background image. To generate these paired images, a collection of 300 fonts and 12,000 background images are used. These background images were subjected to random rotation, curve, and perspective transformations to introduce diversity. It is worth noting that we also generated an additional set of 100,000 labeled images using 2,500 fonts. This expanded dataset was employed specifically for training purposes of the proposed method.

Real Data: In our research on enhancing natural screen text editing on real scene images, we utilized a dataset consisting of real scene images obtained from the MOSTEL [44]. This dataset is generated by the authors using random cropped images from ICDAR 2013 [32], MLT-2017 [38] MLT-2019 [37] datasets.

A. Implementation Details

The implementation process involved training the U-Net backbone using input images resized to 64×256, utilizing a synthetic dataset as outlined in our main submitted draft. *We employ the U-Net to approximate the binary mask of the input image*. The input images are the same as those taken by the CTB Block of our model, and corresponding binary masks are provided in the dataset as ground truth. Our approach utilized the Adam optimizer [33] with a learning rate set at 0.001. The U-Net underwent training for a total of 20 epochs, with each batch containing a single iteration. The complete pipeline was implemented using PyTorch [24] and trained on a single NVIDIA 4080 OC GPU.

B. Further Analysis and Discussions

Not Just Another STE Method Please note that our method is not just "another method" that provides better scene text editing. Our work aims to develop a novel *font-agnostic method* that simultaneously generates text in arbitrary styles and locations while preserving a natural and realistic appearance through a simple combination of mask generation and text style transfer as shown by **attention map visualization** in Figure 9. To the best of our knowledge, this is the first work of its kind. Moreover, our approach differs from the existing methods as they directly modify all image pixels. Instead, the proposed method has introduced a filtering mechanism to remove background distractions, allowing the network to focus solely on the text regions where editing is required as shown in some qualitative examples in Figure 8 for synthetic and Figure 11 for real images. Additionally, our results elegantly spotlight our method's prowess across varied scene text styles. We also see in Figure 10 some examples of images containing arbitrarily shaped text where the model still can perform well.

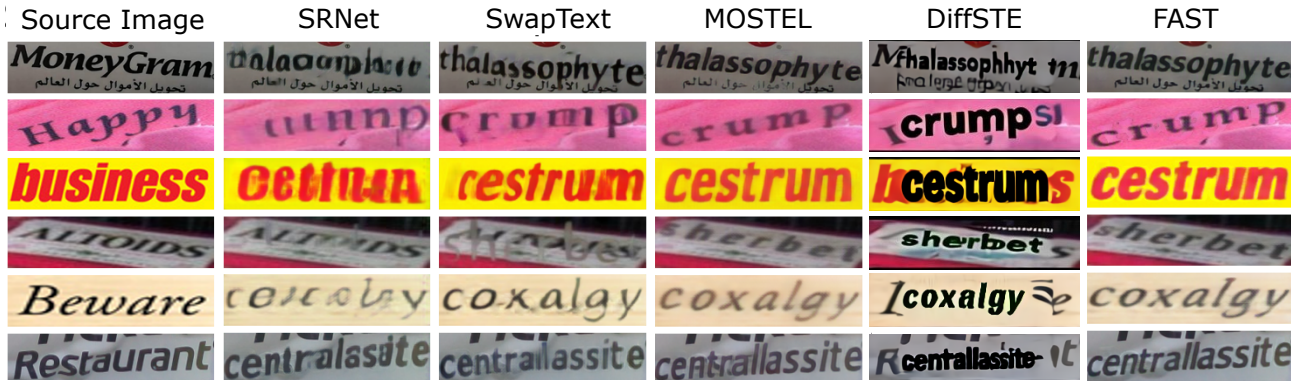

Figure 6. Visual comparison of FASTER with current SOTA methods on real scenes.

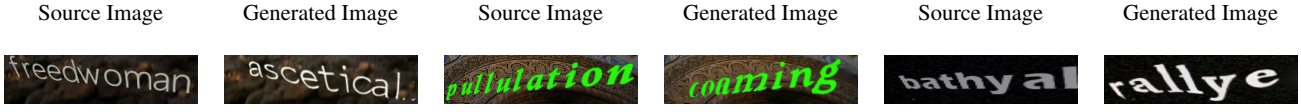

Figure 7. Some failure cases of our model.

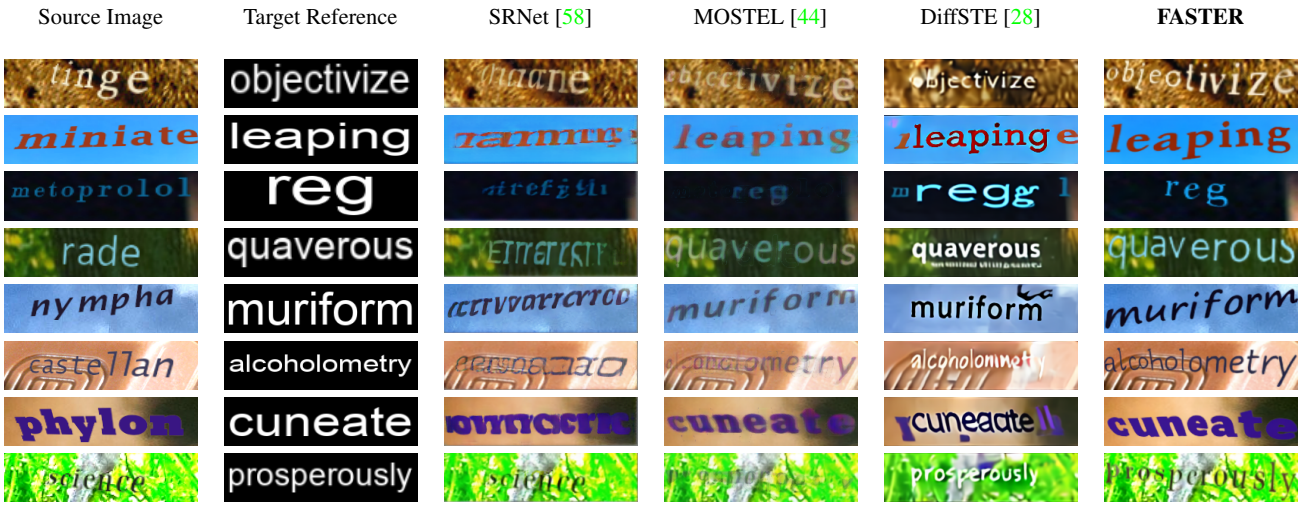

Figure 8. Qualitative comparison of FASTER with SOTA on Synthetic Data samples

Figure 9. **How FASTER Works:** Attention visualization of different phases of image editing using FASTER on a source image with "millrace" and target image with "ferriheme" (from top left to bottom right in order)

Figure 10. Qualitative image editing using FASTER on some arbitrary-shaped text examples from Synthetic dataset

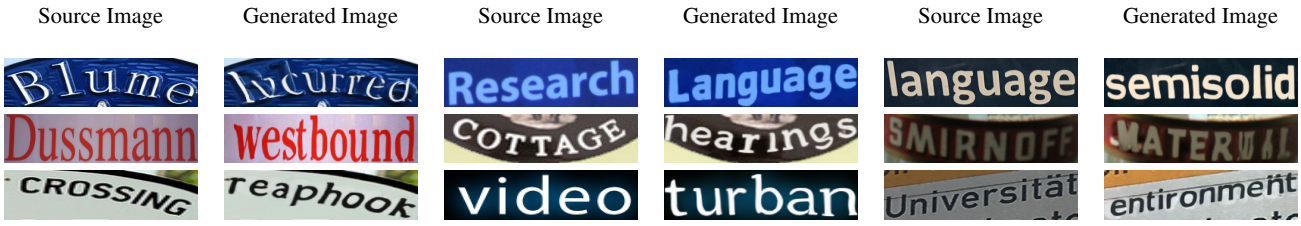

Figure 11. Qualitative image editing examples using FASTER on the Real dataset.

FASTER attains the best results (in terms of average inference time) when compared to existing SOTA STE models. In Table 1, we have thoughtfully presented the average time taken by each of the methods under consideration. This table serves as a valuable reference point for assessing the efficiency of the different approaches. Specifically, it is noteworthy that that "FASTER" stands out in terms of its computational speed during inference. The model *clocks an impressive speed of 31.12 milliseconds for generating a single image*. This efficiency is not only remarkable in isolation but also positions "FASTER" as a significantly swifter option compared to the previously employed methodologies specially *compared to stable diffusion-based [28] approaches which take almost 400 times more time for single image inference*. This observation underscores the strides made in optimizing the computational efficiency of our approach, highlighting its potential to expedite the text image editing operation effectively.

What makes FASTER establish font-agnostic text editing? As demonstrated in Table 6 and Table 7, we examine how different font types and sizes impact our outcomes. This is an ablation study, not a comparison study with SOTA methods and

Figure 12. Qualitative evaluation of editing on real images from Internet

hence incremental accuracy gain is not applicable here. The results from Table 6, where *we explore Arial, Times New Roman, and Sans Serif fonts, indicate minimal influence on our method's performance*. Similarly, Table 7, maintaining *different sizes (25 and 28) of fixed font, also showcases negligible effects.* Importantly, it's worth noting that our training synthetic dataset encompasses diverse fonts and sizes, being a standard OCR dataset. These collective findings underscore our method's font-agnostic nature, where it consistently performs across font variations. It can be noted that if we change different fonts and

font sizes, the accuracy of our method does not change much. Hence, our method is robust in different fonts and font sizes.

C. Visualizing Real-World Scene Editing

This section highlights some of the key observations, qualitative analysis and case studies to demonstrate the potential of FASTER in real-world scene editing applications.

Advertising and Marketing: STE allows marketers to update product details, prices, and promotional messages in real-world advertisements, posters, and billboards efficiently and effectively. As shown in Figure 12, the second example shows an advertisement board, which preserves the background style and structural properties while editing text seamlessly between ”LOWEST” and ”COOLEST”. Also, in third example use-case as shown we see the content edit between ”DTANJUNG” and generated ”DOCKYARD” where the content is in a different language (Malaysian/Indonesian).

Retail and E-commerce: Retailers can update prices, product descriptions, and availability information on shelf labels and in online product images, ensuring accuracy and compliance with marketing strategies. One of the best examples as shown in Figure 14 is the ”DOUBLEMINT” to ”SINGLEMINT” label change with the product brand name. This shows how FASTER can easily impact marketing e-commerce brand visuals.

Artistic and Creative Expressions: As shown in first example in Figure 12, FASTER can mimick and adapt specific font colors seamlessly with content editing. ”HAPPY BIRTHDAY” could integrate ”MERRY BIRTHDAY” preserving all kinds of typographic attributes related to font color, style, letter spacing, alignment and font size. This could eventually help artists and designers use FASTER to incorporate text into their visual creations, with unique and expressive design properties in the user-interface.

Augmented Reality (AR) and Mixed Reality (MR): FASTER enhances the user experience by overlaying informative text, such as navigation instructions or facts, onto real-world scenes in AR and MR applications. As shown in Figure 16, we observe in all the examples how the content could have tampered with the original navigation instructions in response to street signboards or location milestone blocks. It also promotes visual consistency which ensures that text edits align with the overall context and design, promoting an extremely cohesive user experience.

Visually-Rich Document Editing: An interesting case study was performed for some visually-rich document samples as illustrated in Figure 17 and Figure 18. The results illustrate that indeed FASTER has some real potential into content editing for scientific papers as shown in Figure 17. This gives it a really powerful application for editing and generative tasks for document analysis. ”CHAPTER 1” to ”SECTION 1” edit or changing ”2 AUTHORS” to ”2 Members” in the given document image makes it hard for humans to distinguish between real and synthetically edited (generated) documents. Also, Figure 18 shows more infographic-like documents where different header style elements have been preserved with edited content which blends seamlessly with the overall document structure.

Figure 13. More Qualitative Results from real-world Internet images

Figure 14. More Results from Internet Results

Figure 15. Some Specific difficult real-world examples

Figure 16. How FASTER behaves in image editing with stop signals and bulletins in roads

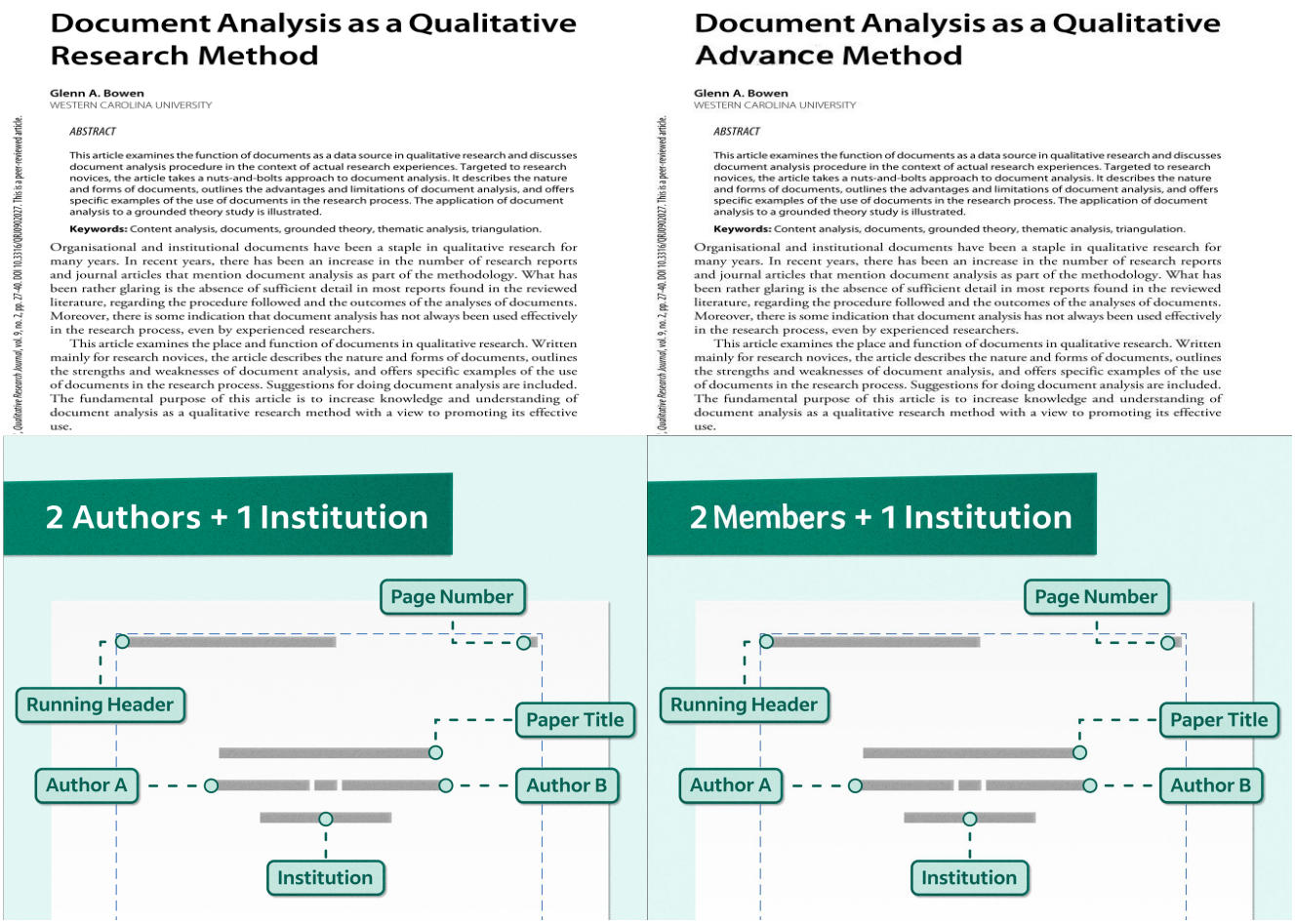

Figure 17. Some results on image editing with FASTER on Document Images

Figure 18. Some more results on image editing with FASTER on Document Images