GlyphSR: A Simple Glyph-Aware Framework for Scene Text Image Super-Resolution

Baole Wei, Yuxuan Zhou, Liangcai Gao*, Zhi Tang

Wangxuan Institute of Computer Technology, Peking University, Beijing, China
weibaole@stu.pku.edu.cn, {sherco, gaoliangcai, tangzhi}@pku.edu.cn

Abstract

The goal of scene text image super-resolution (STISR) is to enhance the clarity of text within line images, thereby improving readability and enabling more accurate text recognition. However, existing STISR methods often rely heavily on Text Prior (TP) derived from trained recognizers, which can be unreliable and may lead to incorrect glyph restoration. Text images contain two crucial types of information: semantic content from word meanings and structural details from glyphs. When semantic information is unreliable, accurate perception of glyph structures becomes essential. This paper introduces GlyphSR, a novel STISR framework that addresses three key challenges: precise extraction, effective learning, and optimal utilization of glyph structural information. GlyphSR incorporates the Glyph Extraction Module (GEM), a training-free approach leveraging the Segment Anything Model (SAM) to accurately extract character-level glyphs. The Glyph Perception Module (GPM) models and learns glyph structures through segmentation and classification tasks, while the Glyph Fusion Module (GFM) integrates glyph information to enhance overall STISR model performance. Extensive experiments on the TextZoom dataset demonstrate that GlyphSR achieves a new state-of-the-art performance.

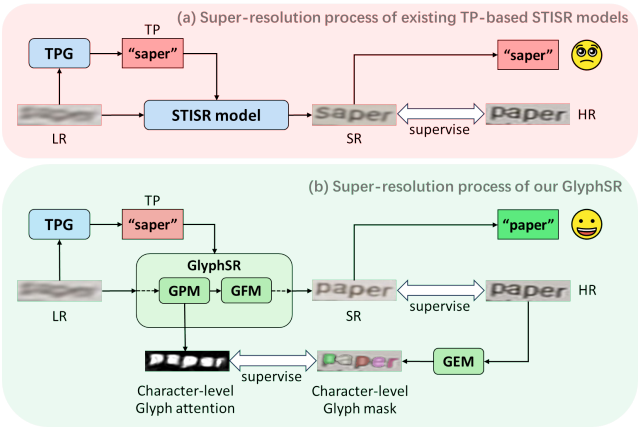

Figure 1: Comparison of the super-resolution process between (a) existing TP-based models and (b) GlyphSR. Most TP-based methods would be misled into a false restoration direction when the TP Generator cannot recognize the characters correctly. In contrast, GlyphSR introduces character-level glyph mask supervision, providing structural information that significantly differs from text prior, thus reducing the risk of the model being misled.

Introduction

Text recognition is the process of transcribing text images into corresponding text sequences. As a crucial step in scene understanding and document image analysis, text recognition has wide-ranging applications in real-world scenarios, such as autonomous driving, assistive technologies for the visually impaired, and book digitization. In recent years, deep learning-based text recognition methods have been increasingly adopted across various industries, achieving satisfactory accuracy under common conditions. However, text images may suffer from degradation during acquisition, transmission, or storage, such as reduced resolution and irreversible information loss due to image compression, or blurring caused by camera shake or low light conditions, resulting in low-resolution (LR) text images, which significantly impair the accuracy of text recognition models. Scene text image super-resolution (STISR) aims to enhance the clarity and readability of LR text images, thereby reducing recognition difficulty and improving the accuracy of subsequent text recognition.

As an image-to-image generation task, STISR requires models to perceive text content (semantic information) effectively and restore text glyphs (structural information) accurately, producing super-resolution (SR) images with improved readability. Most existing STISR approaches incorporate the predicted probability distributions from a trained recognizer as supplementary semantic information, known as Text Prior (TP), to guide the model in restoring text images with an informed understanding of the semantic content in LR images. However, as shown in the upper row of Fig 1, the performance of these methods heavily relies on the quality of the trained recognizer, and unreliable TP can mislead the STISR model, resulting in incorrect glyph structure restoration. Text images contain two critical types of information: semantic information derived from word meanings and structural information derived from glyphs. Per-

*Corresponding author.

ception of glyph information is crucial when semantic information is unreliable. TBSRN(Chen, Li, and Xue 2021) and Text Gestalt(Chen et al. 2021) introduce alignment mechanisms through the attention scores of pre-trained recognizers to guide the model's focus on character regions; C3-STISR(Zhao et al. 2022) renders standard glyph text line images based on TP as an additional structural prior. However, these methods still fall short in accurately modeling glyph structural information. Attention scores may fail to accurately represent glyph structures, and standard glyph text line images can differ significantly from actual glyphs.

Therefore, the objective of this work is to construct and fully utilize reliable glyph structural information guidance. We identify three key challenges that need to be addressed: (1) how to accurately extract glyph information, (2) how to effectively learn glyph information, and (3) how to make full use of glyph information. To address these challenges, we propose a straightforward and effective glyph-aware STISR framework, GlyphSR. Specifically, as shown in the lower row of Fig 1, to address the first challenge, we design the training-free **G**lyph **E**xtraction **M**odule (GEM), leveraging the powerful local structure segmentation capabilities of Segment Anything Model (SAM)(Kirillov et al. 2023) to accurately extract glyph-level images. To tackle the second challenge, we introduce the **G**lyph **P**erception **M**odule (GPM) that explicitly models and learns glyph structural information through three tasks: character glyph segmentation, text line glyph segmentation, and character classification. Finally, to overcome the third challenge, we propose a glyph-based feature enhancement module, **G**lyph **F**usion **M**odule (GFM), which uses the learned glyph information to enhance the intermediate features of the STISR model, thereby improving its glyph-awareness and ultimately enhancing text image restoration quality. Our contributions can be summarized as follows:

- We propose a simple yet effective training-free glyph extraction method, Glyph Extraction Module, which accurately extracts character-level glyph images without requiring additional training data.
- We introduce the Glyph Perception Module and the Glyph Fusion Module, which together enable effective modeling, learning, and utilization of glyph structural information.
- Extensive experiments are constructed on the TextZoom dataset, demonstrating the effectiveness of our approach, achieving a new state-of-the-art performance.

Related Works

Scene Text Recognition

Scene Text Recognition (STR) focuses on reading text from natural images. CRNN (Shi, Bai, and Yao 2015) was the first to introduce a CNN+LSTM framework for sequential text classification, addressing alignment issues with CTC loss. ASTER (Shi et al. 2019) adapted the Seq2Seq framework from natural language processing, using an RNN decoder for recognition and an STN (Jaderberg et al. 2015) to handle irregular text arrangements. SRN (Yu et al. 2020) replaced the RNN with a Transformer, leveraging global attention for parallel character prediction. ABINet (Fang et al. 2021) introduced bidirectional feature representation with a language model for result correction, while PARSeq (Bautista and Atienza 2022) employed permutation language modeling for improved performance. CA-FCN (Liao et al. 2019) used segmentation to locate and classify character regions, and SIGA (Guan et al. 2023) achieved comparable results without a language model by incorporating self-supervised implicit glyph attention. CAM (Yang et al. 2024) employed class-aware glyph masks and a feature alignment-fusion module to refine features and suppress background noise, enhancing scene text recognition performance.

Scene Text Image Super-Resolution

Early deep learning-based STISR methods relied on CNNs. The introduction of GANs (Goodfellow et al. 2014) led to adversarial training in TextSR (Wang et al. 2019), enhancing focus on text content. Plugnet (Mou et al. 2020) used multi-task learning for simultaneous super-resolution and recognition, unifying features for both tasks.

TextZoom (Wang et al. 2020), featuring real-world image pairs with varying resolutions, marked a significant milestone. TSRN (Wang et al. 2020) employed bidirectional LSTMs to utilize sequential information both horizontally and vertically. Building on TSRN, TBSRN (Chen, Li, and Xue 2021) incorporated attention-based position-aware and content-aware losses, while TPGSR (Ma, Guo, and Zhang 2021) used text priors from recognition to guide super-resolution. TG (Chen et al. 2021) leveraged stroke-level annotations for fine-grained cues, and TATT (Ma, Liang, and Zhang 2022) introduced Structure Consistency Loss for irregular text. C3-STISR (Zhao et al. 2022) combined recognition, visual, and linguistic information for improved guidance. DPMN (Zhu et al. 2023a) provided a plug-and-play modulation module to enhance existing networks. LEMMA (Guo et al. 2023) used attention to extract character regions for higher-level supervision.

RGDiffSR (Zhou et al. 2023) and TextDiff (Liu et al. 2023) integrated recognizer guidance into diffusion-based restoration, with TextDiff further refining super-resolution results. PEAN (Zhao et al. 2024) restored text priors via a diffusion model and used them to guide the super-resolution process. These diffusion-based methods have achieved superior semantic accuracy.

Text Segmentation

Text segmentation aims to separate each character from the background by identifying its boundaries. Deep learning methods have significantly outperformed traditional threshold-based approaches. For example, TexRNet (Xu et al. 2020) uses a CNN tailored for text region segmentation in natural images, while EAformer (Yu et al. 2024) combines a Transformer with a text edge extractor to further enhance accuracy. (Wang et al. 2022) proposed a self-supervised framework that uses polygon-level masks from a text localization network as input, enabling segmentation without pixel-level supervision. However, most existing algorithms focus on segmenting entire text regions, typically

providing only line-level segmentation masks and lacking the capability to segment individual characters.

Recently, the Segment Anything Model (SAM) (Kirillov et al. 2023; Ravi et al. 2024), a large-scale pre-trained model, has shown exceptional performance across various segmentation tasks. With its support for prompt-based output, SAM can generate customized segmentation results for diverse downstream tasks. For instance, SAM-Track (Cheng et al. 2023) uses SAM for object tracking, MedSAM (Ma et al. 2024) for medical image segmentation, and Hi-SAM (Ye et al. 2024) for text segmentation and layout analysis. We leverage SAM, using text localization information as prompts to obtain character-level glyph segmentation masks.

Method

Overall Architecture

Our proposed GlyphSR framework comprises five main modules: TP Generator (TPG), SR Branch, Glyph Extraction Module (GEM), Glyph Perception Module (GPM), and Glyph Fusion Module (GFM).

Both the TPG and SR Branch follow standard TP-based model structures. As shown in Fig 2, given an input LR image $I_{lr} \in \mathbb{R}^{H\times W\times C_{img}}$, it is first processed by the TPG, which includes a trained text recognizer and a TP transformer. The predicted probability distribution from the recognizer is mapped by the TP transformer using a deconvolution structure to form $TP \in \mathbb{R}^{H\times W\times C_{tp}}$.

In the SR Branch, I_{lr} is passed through a shallow convolutional layer to extract visual features, followed by a series of SR blocks that produce intermediate features $F_{sr}^{n} \in \mathbb{R}^{H\times W\times C_{sr}}$, where $n = 1, 2, 3, \ldots, N$ and N is the number of SR blocks. The TP is fed into each SR block as semantic guidance.

The GEM generates pseudo-labels for line-level glyphs $\hat{M}_{line} \in \mathbb{R}^{H\times W\times 1}$ and character-level glyphs $\hat{M}_{char} \in \mathbb{R}^{H\times W\times L}$ from the high-resolution (HR) image $I_{hr} \in \mathbb{R}^{2H\times 2W\times C_{img}}$, where L represents the number of characters in the text image.

The GPM uses $F_{sr}^{N_{gly}}$ to predict the line-level glyph map $M_{line} \in \mathbb{R}^{H\times W\times 1}$ and the character-level glyph map $M_{char} \in \mathbb{R}^{H\times W\times L}$, where N_{gly} denotes the index of the intermediate feature used as input to the GPM.

In the GFM, M_{char} is fused with $F_{sr}^{n_{enh}}$ to produce the glyph-enhanced feature $F_{gly}^{n_{enh}} \in \mathbb{R}^{H\times W\times C_{sr}}$, where $n_{enh} = N_{gly}, N_{gly}+1, \ldots, N$. The final glyph-enhanced feature F_{gly}^{N} is then fed into the upsampling and output layers in the SR Branch to generate the super-resolution (SR) image $I_{sr} \in \mathbb{R}^{2H\times 2W\times C_{img}}$.

The core contributions of this work lie in the GEM, GPM, and GFM. To maintain simplicity while ensuring robust performance, the SR Branch follows the standard TPGSR(Ma, Guo, and Zhang 2021) implementation, with the recognizer in the TPG chosen as PARSeq(Bautista and Atienza 2022), similar to PEAN(Zhao et al. 2024), and the TP transformer using the deconvolution and deformable convolution structure from C3-STISR(Zhao et al. 2022).

Glyph Extraction Module

Previous STISR works have attempted to enhance character-awareness by aligning attention maps from trained recognizers. However, attention maps only approximate character locations and are unable to capture detailed glyph structures. Additionally, they are prone to drift in complex scenes, as the primary goal of attention-based recognizers is text transcription, not precise structural extraction.

The Segment Anything Model (SAM) (Kirillov et al. 2023), a widely adopted visual foundation model, excels at local structure extraction and generalization. With appropriate guidance, SAM can perform effectively across various downstream tasks. Here, we introduce the Glyph Extraction Module (GEM), a training-free method for character-level glyph extraction, guiding SAM to extract glyph structures using reliable prompt points.

Ideally, each prompt point input would yield a corresponding glyph segmentation mask for a single character. However, selecting effective prompt points is challenging. To address this, we first extract the center of each character as a key point, then expand around it to generate anchor points, which are used as prompt points for SAM. GEM primarily consists of two steps: key point extraction and anchor-based mask selection.

Key Point Extraction We utilize two types of attention to extract two key points per character:

2D Position Attention: This attention map is obtained from a Transformer-based recognition model pre-trained on synthetic datasets like Syn90k(Gupta, Vedaldi, and Zisserman 2016) and SynthText(Gupta, Vedaldi, and Zisserman 2016). The attention map at each time step t indicates the approximate position of the t-th character. With extensive pre-training, this module reliably predicts character positions but only approximates their horizontal and vertical locations. Since the 2D Position Attention map typically radiates from the center, we extract the center as the key point. We first apply morphological filtering to remove low-response points from high-response regions, followed by connected component analysis to isolate the center, serving as the first key point.

Glyph Attention: Following SIGA (Guan et al. 2023), we perform foreground-background segmentation using a pre-trained self-supervised CNN, with labels obtained via K-means clustering. We also extract 1D attention during inference using an RNN-based classifier for horizontal position cues. The element-wise multiplication of the foreground-background segmentation and the expanded 2D attention map yields the Glyph Attention map. Although more precise, the Glyph Attention map is irregular, so we extract its minimum bounding rectangle and use the center as the second key point.

Anchor-based Mask Selection To enhance segmentation robustness with SAM, we adopt an anchor strategy from object detection. We shift the key point in four directions (up, down, left, right) with K different step sizes, generating $4K + 1$ anchor points per key point. These anchors serve as positive point prompts for SAM to generate the corresponding character mask.

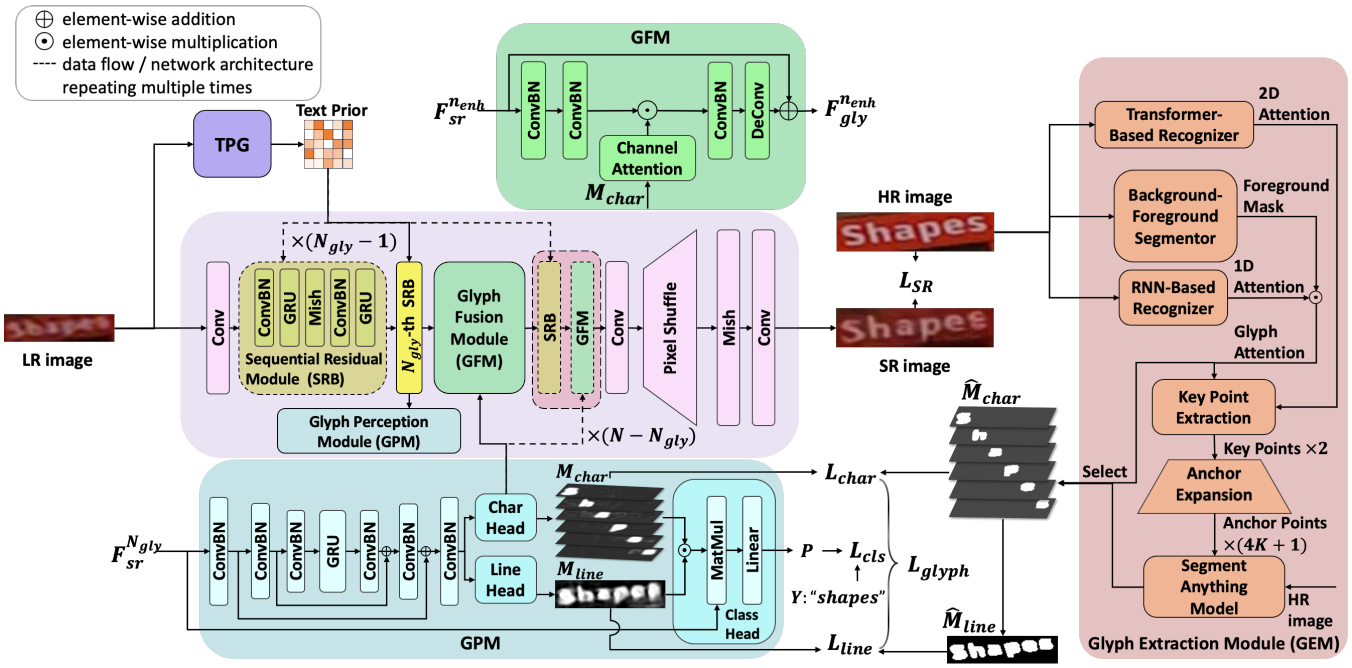

Figure 2: The architecture of our GlyphSR framework.

After generating candidate masks for all anchor points, we select the final mask based on two criteria: SAM's confidence score and the Intersection over Union (IoU) between the mask and Glyph Attention regions. We first filter masks with confidence scores above a threshold τ, then select the mask with the highest IoU. In cases where Glyph Attention is unreliable (e.g., empty attention maps or incorrect segmentation), we rank masks by $\frac{confidence\ score}{\sqrt{area(mask)}}$ and select the highest-scoring mask, where $area(\cdot)$ represents the number of positive pixels in the binarized mask. This process ensures accurate glyph mask extraction for each character.

Glyph Perception Module

Building on the extracted glyph information from the GEM, the Glyph Perception Module (GPM) is designed to effectively learn from the glyph pseudo-labels $\hat{M}_{line}$ and $\hat{M}_{char}$. As shown in Fig 2, given the intermediate features $F_{sr}^{N_{gly}}$, they are first passed through a three-layer pyramid-like multi-scale feature extraction structure, where each level of the downsampling stage is added to the corresponding level of the upsampling stage. To achieve sequential modeling, the end-level features of the downsampling stage are encoded by a GRU layer. After the upsampling stage, three sets of convolutional structures are introduced to handle different learning tasks: line-level glyph segmentation, character-level glyph segmentation, and character classification. Finally, the three prediction heads are supervised by the glyph pseudo-labels from the GEM and the text labels, as detailed below:

In the line-level glyph segmentation head (Line Head), given the output features from the multi-scale feature extraction structure, a single convolutional layer is used to predict the line-level glyph segmentation map $M_{line} \in \mathbb{R}^{H\times W\times 1}$. We use MSE loss to calculate the difference between M_{line} and $\hat{M}_{line}$, as shown in the following equation:

$$L_{line} = \left\| M_{line} - \hat{M}_{line} \right\|_2, \tag{1}$$

Similarly, the output $M_{char} \in \mathbb{R}^{H\times W\times L}$ of the character-level glyph segmentation head (Char Head) is supervised by $\hat{M}_{char}$, using MSE loss as follows:

$$L_{char} = \left\| M_{char} - \hat{M}_{char} \right\|_2, \tag{2}$$

To further guide the model in learning the alignment between glyph structures and character semantics, we introduce a character classification head (Class Head). We treat M_{line} and M_{char} as line-level and character-level attention scores, respectively, and rearrange the intermediate features $F_{sr}^{N_{gly}}$ to obtain character sequence features, and use a linear layer to predict character class logits $P \in \mathbb{R}^{L\times D}$, where D represents the number of character classes. Supervision is applied using text labels $Y \in \mathbb{R}^{L}$ and cross-entropy loss as expressed in the following equations:

$$\begin{aligned} Attn &= M_{char} \odot M_{line}, \\ P &= Linear(Attn \cdot F_{sr}^{N_{gly}}), \\ L_{cls} &= CE(P, Y), \end{aligned} \tag{3}$$

Finally, the losses from the three prediction heads are combined to form the loss function for the GPM:

$$L_{glyph} = L_{line} + L_{char} + L_{cls}. \tag{4}$$

Figure 3: Qualitative comparison with previous methods.

Glyph Fusion Module

In the GPM, we introduced glyph-related learning tasks to guide the model in learning to extract glyph information. Building on this, we aim to make better use of the glyph information learned by the GPM. In this subsection, we propose the Glyph Fusion Module (GFM), which explicitly fuses the character-level glyph maps M_{char} predicted by the GPM with the intermediate features $F_{sr}^{n_{enh}}$, further enhancing the model's glyph perception capability.

Since the channel dimension of M_{char} represents the character sequence, which significantly differs from that of $F_{sr}^{n_{enh}}$, as shown in Fig 2, the GFM first applies two convolutional layers to $F_{sr}^{n_{enh}}$ for channel mapping, obtaining the hidden features $F_{hid0}^{n_{enh}}$ with the same number of channels as M_{char}. Next, M_{char} undergoes channel-wise attention rearrangement to obtain M'_{char}, reducing the impact of padding positions in the text sequence. Then, $F_{hid0}^{n_{enh}}$ is element-wise multiplied by M'_{char} and passed through a convolutional layer to map the channel dimension back to that of the $F_{sr}^{n_{enh}}$. Subsequently, the hidden features are input into a deformable convolution layer to obtain the enhanced hidden features $F_{hid1}^{n_{enh}}$, aligning it spatially with $F_{sr}^{n_{enh}}$. Finally, the original intermediate features $F_{sr}^{n_{enh}}$ are added as a residual to $F_{hid1}^{n_{enh}}$, and after ReLU activation, the glyph-enhanced intermediate features $F_{gly}^{n_{enh}}$ are obtained, serving as the input for the next block in the SR branch. The module is formulated as follows:

$$\begin{aligned} F_{hid0}^{n_{enh}} &= Conv1(Conv0(F_{sr}^{n_{enh}})), \\ M'_{char} &= ChannelWiseAttention(M_{char}), \\ F_{hid1}^{n_{enh}} &= DeformConv(Conv2(F_{hid0}^{n_{enh}} \odot M'_{char})), \\ F_{gly}^{n_{enh}} &= ReLU(F_{hid1}^{n_{enh}} + F_{sr}^{n_{enh}}). \end{aligned} \tag{5}$$

Optimization

The training loss function of GlyphSR consists of two parts: the SR loss L_{sr} and the Glyph loss L_{glyph} introduced in the GPM. The SR loss follows the common loss functions used in mainstream STISR methods, namely pixel loss, text focus loss, and HOG loss, as formulated below:

$$\begin{aligned} L_{pix} &= \|I_{sr} - I_{hr}\|_2, \\ L_{tf} &= \|A_{sr} - A_{hr}\|_1 + WCE(P_{sr}, Y), \\ L_{hog} &= \|HOG(I_{sr}) - HOG(I_{hr})\|_1, \\ L_{sr} &= \lambda_1 L_{pix} + \lambda_2 L_{tf} + \lambda_3 L_{hog}, \end{aligned} \tag{6}$$

where I_{sr} and I_{hr} represent the predicted SR image and the HR image, respectively. $A(\cdot)$ and WCE denote the attention map from a pre-trained transformer-based recognizer and the weighted cross-entropy loss, respectively, and $HOG(\cdot)$ represents the HOG features of the corresponding input. The $\lambda_1, \lambda_2, \lambda_3$ are set to $1, 1, 0.1$.

Finally, the overall loss function of GlyphSR is formulated as:

$$L = L_{sr} + L_{glyph}. \tag{7}$$

Experiments

Datasets and Evaluation Metric

Following mainstream STISR works, we conduct experiments on the TextZoom(Wang et al. 2020) dataset. TextZoom is captured in real-world scenarios using digital cameras. It contains 17,367 LR-HR image pairs for training and 4,373 image pairs for testing. The test set is divided into three subsets: the easy subset with 1,619 image pairs, the medium subset with 1,411 image pairs, and the hard subset with 1,343 image pairs. The LR images have a size of 16×64, while the HR images have a size of 32×128.

We use the text recognition accuracy as the main evaluation metric, consistent with previous work. Three text recognizers are adopted for evaluation, namely CRNN(Shi, Bai, and Yao 2015), MORAN(Luo, Jin, and Sun 2019) and ASTER(Shi et al. 2019). In our ablation studies, we use Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) to assess the quality of the generated SR images.

Method	CRNN(Shi, Bai, and Yao 2015)				MORAN(Luo, Jin, and Sun 2019)				ASTER(Shi et al. 2019)			
	easy	medium	hard	average	easy	medium	hard	average	easy	medium	hard	average
Bicubic	36.4%	21.1%	21.1%	26.8%	60.6%	37.9%	30.8%	44.1%	64.7%	42.4%	31.2%	47.2%
TG(Chen et al. 2021)	61.2%	47.6%	35.5%	48.9%	75.8%	57.8%	41.4%	59.4%	77.9%	60.2%	42.4%	61.3%
TATT(Ma, Liang, and Zhang 2022)	62.6%	53.4%	39.8%	52.6%	72.5%	60.2%	43.1%	59.5%	78.9%	63.4%	45.4%	63.6%
C3-STISR(Zhao et al. 2022)	65.2%	53.6%	39.8%	53.7%	74.2%	61.0%	43.2%	60.5%	79.1%	63.3%	46.8%	64.1%
DPMN(+TATT)(Zhu et al. 2023a)	64.4%	54.2%	39.2%	53.4%	73.3%	61.5%	43.9%	60.4%	79.3%	64.1%	45.2%	63.9%
TSAN($Zhuet\ al.$ 2023b)	64.6%	53.3%	38.8%	53.0%	78.4%	61.3%	45.1%	62.7%	79.6%	64.1%	45.3%	64.1%
LEMMA($Guoet\ al.$ 2023)	67.1%	58.8%	40.6%	56.3%	77.7%	64.6%	44.6%	63.2%	81.1%	66.3%	47.4%	66.0%
RTSRN($Zhanget\ al.$ 2023)	67.0%	59.2%	42.6%	57.0%	77.1%	63.3%	46.5%	63.2%	80.4%	66.1%	49.1%	66.2%
TextDiff($Liuet\ al.$ 2023)	64.8%	55.4%	39.9%	54.2%	77.7%	62.5%	44.6%	62.7%	80.8%	66.5%	48.7%	66.4%
RGDiffSR($Zhouet\ al.$ 2023)	67.6%	56.5%	42.7%	56.4%	78.6%	62.1%	45.4%	63.1%	81.1%	65.4%	49.1%	66.2%
PEAN($Zhaoet\ al.$ 2024)	68.9%	60.2%	45.9%	59.0%	79.4%	67.0%	49.1%	66.1%	84.5%	**71.4**%	52.9%	70.6%
GlyphSR (Ours)	**71.4**%	**64.8**%	**48.0**%	**62.1**%	**82.0**%	**69.0**%	**51.5%**	**68.5**%	**84.7**%	71.1%	**54.3**%	**71.0**%
HR	76.4%	75.1%	64.6%	72.4%	91.2%	85.3%	74.2%	84.1%	94.2%	87.7%	76.2%	86.6%

Table 1: Comparison of downstream text recognition accuracy with other SOTA methods on TextZoom dataset. Bolded numbers denote the best results.

Implementation Details

The number of SRBs N is set to 5, and and the channel sizes are $C_{sr} = 64$ and $C_{tp} = 32$. The maximum text length L is 15. The optimal SRB index for GPM and GFM N_{gly} is 4, which we will demonstrate in the ablation study.

For the GEM, we employed the transformer-based recognizer from (Chen, Li, and Xue 2021), and the RNN-based classifier from (Guan et al. 2023). When generating anchor points, the number of step sizes is set to $K = 2$, with corresponding step sizes $\{2, 4\}$. The confidence score threshold τ is set to 0.8 for mask selection.

We use Adam optimizer for training. The batch size is set to 48. The number of training epochs is set to 200. The learning rate is initialized to 0.001, and CosineAnnealingLR with 5 epochs linear warmup is used to adjust it. All our experiments are conducted with a single NVIDIA Tesla A800 GPU for both training and testing.

Comparison to State-of-the-Art Methods

We validate our proposed GlyphSR model on the TextZoom dataset and compare it with previous state-of-the-art (SOTA) models. As shown in Table 1, GlyphSR achieves improvements across almost all recognizers and test set configurations, demonstrating the superiority and effectiveness of our proposed method. Specifically, GlyphSR improves the average accuracy by 3.1% in tests with the CRNN recognizer, by 2.4% with the MORAN recognizer, and by 0.4% with the ASTER recognizer.

Although GlyphSR shows limited improvement on the easy and medium subsets with the ASTER recognizer, it achieves a 1.4% gain on the hard subset. This evidence suggests that incorporating glyph structural information can significantly enhance the model's robustness in complex scenes.

As illustrated in Fig 3, we also conduct a qualitative comparison on a series of typical cases, all of which are sourced from the hard subset. Observation shows that these LR images suffer from character distortion and adhesion due to blurring, leading to high character ambiguity and low reliability of text recognition. Existing TP-based methods are prone to errors, as they are easily misled by unreliable TP. In contrast, GlyphSR, with its enhanced glyph-awareness, is capable of high-quality restoration even in cases of low TP reliability. This further confirms the importance of glyph information for the STISR task and validates the effectiveness of our proposed method.

Ablation Study

In this subsection, we conduct ablation studies on the three core modules of GlyphSR, including Glyph Extraction Module (GEM), Glyph Perception Module (GPM), and Glyph Fusion Module (GFM).

Method	Recognition Accuracy				Image Fidelity	
	easy	medium	hard	average	PSNR	SSIM
Baseline1	67.76%	60.38%	45.94%	58.68%	20.68	0.7535
Baseline2	69.49%	61.16%	46.69%	59.80%	<u>21.07</u>	<u>0.7618</u>
w/o GFM	<u>70.78%</u>	<u>62.08%</u>	**48.10**%	<u>61.01%</u>	**21.33**	**0.7819**
GlyphSR	**71.40**%	**64.78**%	<u>48.03%</u>	**62.09**%	20.74	0.7591

Table 2: Ablation study on GEM and GFM. The best results are highlighted in bold, and the second-best results are underlined.

Effectiveness of Glyph Extraction Module To verify the effectiveness of GEM, we construct two baseline models. Baseline1 retains only the TPG and the SR branch of GlyphSR and is trained using only the SR loss, without any additional learning tasks. In Baseline2, we retain the GPM structure and use the foreground-background clustering results based on K-means, as proposed in (Guan et al. 2023), as row-level glyph pseudo-labels for model training. For fair comparison, we use GlyphSR without GFM (w/o GFM) to evaluate text recognition and image quality metrics. As shown in Table 2, Baseline2 outperforms Baseline1 due to the additional glyph information. Our w/o GFM model consistently surpasses both baselines, confirming that the glyph information extracted by GEM is more reliable than existing methods. Adding GFM further improves recognition accuracy, demonstrating its effectiveness, though it slightly reduces image fidelity. This decline may occur because GFM focuses on enhancing text regions at the expense of background quality. Nevertheless, w/o GFM still outperforms the baselines in image fidelity, affirming the robustness of GEM. Detailed analysis is provided in Appendix A.

	Tasks			Recognition Accuracy			
	LingSeg	CharSeg	CharCls	easy	medium	hard	average
1				67.76%	60.38%	45.94%	58.68%
2	✓			70.72%	61.37%	46.61%	60.30%
3		✓		70.41%	62.23%	46.24%	60.34%
4			✓	69.86%	61.87%	46.61%	60.14%
5	✓	✓		70.91%	61.94%	47.13%	60.71%
6	✓		✓	70.48%	61.59%	48.10%	60.74%
7		✓	✓	70.41%	61.37%	46.98%	60.30%
8	✓	✓	✓	70.78%	62.08%	48.10%	61.01%

Table 3: Ablation study on GPM.

Effectiveness of Glyph Perception Module As shown in Table 3, we conducted a comprehensive comparison experiment with $2^3 = 8$ groups targeting the three learning tasks in GPM: Line-level glyph Segmentation (LineSeg), Character-level glyph Segmentation (CharSeg), and Character Classification (CharCls). The experimental results confirm that the supervision of each learning task is effective. By comparing the first four groups of experiments—i.e., the baseline model and models that individually learn the LineSeg, CharSeg, and CharCls tasks within GPM—we can see that adding any single learning task supervision significantly improves recognition accuracy. Further, by comparing the last four groups—i.e., the full GlyphSR model with models that omit the LineSeg, CharSeg, or CharCls tasks within GPM—we observe that removing any single learning task results in a decline in recognition accuracy. Notably, comparing the 4th and 5th rows, the model in the 5th row, which only learns the glyph-related LineSeg and CharSeg tasks, outperforms the model in the 4th row, which only learns the text recognition-related CharCls task. This indicates that the glyph information learned by the model is sufficient to represent the semantic information corresponding to character categories, and even benefits from the additional structural information, leading to improved performance. This validates both the reliability of the glyph information extracted by GEM and the effectiveness of the glyph information learning in GPM.

N_{gly}	Recognition Accuracy			
	easy	medium	hard	average
None	70.78%	62.08%	48.10%	61.01%
3	65.47%	58.33%	43.56%	56.44%
4	71.40%	64.78%	48.03%	62.09%
5	70.04%	64.49%	48.40%	61.60%

Table 4: Ablation study on GFM.

Effectiveness of Glyph Fusion Module In this part, we explore the timing and impact of integrating GFM. As shown in Table 4, we evaluate the CRNN recognition accuracy of models that incorporate GFM starting from the 3rd, 4th, or 5th SRB, as well as a model that does not include GFM. The results indicate that the model integrating GFM from the 4th SRB achieves the highest recognition accuracy, followed by the model starting from the 5th SRB, while the model starting from the 3rd SRB shows significantly lower accuracy. This is due to two factors: on one hand, if GFM is introduced too early, the SRBs may not yet have extracted critical deep features, and fusing the backbone features with glyph features at this stage might cause the model to focus on incorrect features. On the other hand, if GFM is introduced too late, the backbone and glyph features may not fully integrate, preventing the model from fully leveraging the informative glyph features, resulting in suboptimal recognition accuracy. Therefore, incorporating GFM at the right stage provides a stable improvement while ensuring the model's glyph learning capability. The model that integrates GFM after the 4th SRB achieves the highest recognition accuracy in this experiment. Under the CRNN TPG setting, our method still achieves consistent improvements, as detailed in Appendix B.

Discussion

As shown in Fig 1 and 3, our proposed GlyphSR achieves outstanding performance in restoring highly distorted scene text images, with the character-level glyph masks extracted by GEM making a significant contribution. However, as illustrated in Fig 2, the masks for characters like 'a' and 'e' sometimes incorrectly fill the hollow regions, marking them as positive areas. This issue likely stems from two factors: (1) For characters with hollow centers, our key point extraction method tends to select these central regions, inadvertently including them in the glyph masks; (2) SAM may struggle with segmenting small-scale text images. Despite these limitations, the glyph masks extracted by GEM and predicted by GPM reliably capture character positions and outer contours, which are crucial for text image restoration. Detailed visualizations are provided in Appendix C. In future work, we plan to fine-tune SAM using negative point prompts and recognition-related objectives to enhance its performance on text images. Furthermore, we aim to explore improved key point selection methods.

Conclusion

In this paper, we presented GlyphSR, a simple yet effective framework for scene text image super-resolution (STISR) that addresses the key challenges of extracting, learning, and utilizing glyph structural information. We introduced the Glyph Extraction Module (GEM), which leverages SAM's powerful segmentation capabilities to accurately extract character-level glyph images without additional training. To effectively learn and model glyph structural information, we proposed the Glyph Perception Module (GPM), which explicitly focuses on character glyph segmentation, text line glyph segmentation, and character class prediction. Furthermore, we developed the Glyph Fusion Module (GFM), which integrates the learned glyph information into the intermediate features of the STISR model, enhancing glyph-awareness and ultimately improving text image restoration quality. Extensive experiments on the TextZoom dataset validated the effectiveness of our approach, demonstrating that GlyphSR sets a new state-of-the-art performance in STISR tasks.

Acknowledgments

This work is supported by the projects of National Natural Science Foundation of China (No. 62376012) and Beijing Science and Technology Program (Z231100007423011), which is also a research achievement of State Key Laboratory of Multimedia Information Processing and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

References

Bautista, D.; and Atienza, R. 2022. Scene Text Recognition with Permuted Autoregressive Sequence Models. In *Computer Vision – ECCV 2022*, 178–196. Cham: Springer Nature Switzerland. ISBN 978-3-031-19815-1.

Chen, J.; Li, B.; and Xue, X. 2021. Scene Text Telescope: Text-Focused Scene Image Super-Resolution. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12021–12030.

Chen, J.; Yu, H.; Ma, J.; Li, B.; and Xue, X. 2021. Text Gestalt: Stroke-Aware Scene Text Image Super-Resolution. *CoRR*, abs/2112.08171.

Cheng, Y.; Li, L.; Xu, Y.; Li, X.; Yang, Z.; Wang, W.; and Yang, Y. 2023. Segment and Track Anything. arXiv:2305.06558.

Fang, S.; Xie, H.; Wang, Y.; Mao, Z.; and Zhang, Y. 2021. Read Like Humans: Autonomous, Bidirectional and Iterative Language Modeling for Scene Text Recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7094–7103. IEEE Computer Society.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Networks. arXiv:1406.2661.

Guan, T.; Gu, C.; Tu, J.; Yang, X.; Feng, Q.; Zhao, Y.; and Shen, W. 2023. Self-Supervised Implicit Glyph Attention for Text Recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15285–15294.

Guo, H.; Dai, T.; Meng, G.; and Xia, S.-T. 2023. Towards robust scene text image super-resolution via explicit location enhancement. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23. ISBN 978-1-956792-03-4.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic Data for Text Localisation in Natural Images. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2315–2324.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; and kavukcuoglu, k. 2015. Spatial Transformer Networks. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. arXiv:2304.02643.

Liao, M.; Zhang, J.; Wan, Z.; Xie, F.; Liang, J.; Lyu, P.; Yao, C.; and Bai, X. 2019. Scene text recognition from two-dimensional perspective. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press. ISBN 978-1-57735-809-1.

Liu, B.; Yang, Z.; Wang, P.; Zhou, J.; Liu, Z.; Song, Z.; Liu, Y.; and Xiong, Y. 2023. TextDiff: Mask-Guided Residual Diffusion Models for Scene Text Image Super-Resolution. arXiv:2308.06743.

Luo, C.; Jin, L.; and Sun, Z. 2019. A Multi-Object Rectified Attention Network for Scene Text Recognition. *CoRR*, abs/1901.03003.

Ma, J.; Guo, S.; and Zhang, L. 2021. Text Prior Guided Scene Text Image Super-resolution. *ArXiv*, abs/2106.15368.

Ma, J.; Kim, S.; Li, F.; Baharoon, M.; Asakereh, R.; Lyu, H.; and Wang, B. 2024. Segment Anything in Medical Images and Videos: Benchmark and Deployment. arXiv:2408.03322.

Ma, J.; Liang, Z.; and Zhang, L. 2022. A Text Attention Network for Spatial Deformation Robust Scene Text Image Super-resolution. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mou, Y.; Tan, L.; Yang, H.; Chen, J.; Liu, L.; Yan, R.; and Huang, Y. 2020. PlugNet: Degradation Aware Scene Text Recognition Supervised by a Pluggable Super-Resolution Unit. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV*, 158–174. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-030-58554-9.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. 2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Shi, B.; Bai, X.; and Yao, C. 2015. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *CoRR*, abs/1507.05717.

Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2019. ASTER: An Attentional Scene Text Recognizer with Flexible Rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9): 2035–2048.

Wang, W.; Xie, E.; Liu, X.; Wang, W.; Liang, D.; Shen, C.; and Bai, X. 2020. Scene Text Image Super-Resolution in the Wild. In *Computer Vision – ECCV 2020*, 650–666. Cham: Springer International Publishing. ISBN 978-3-030-58607-2.

Wang, W.; Xie, E.; Sun, P.; Wang, W.; Tian, L.; Shen, C.; and Luo, P. 2019. TextSR: Content-Aware Text Super-Resolution Guided by Recognition. *CoRR*, abs/1909.07113.

Wang, Y.; Ye, Y.; Mao, Y.; Yu, Y.; and Song, Y. 2022. Self-supervised Scene Text Segmentation with Object-centric Layered Representations Augmented by Text Regions. In *Proceedings of the 30th ACM International Conference on*

Multimedia, MM '22, 5980–5989. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392037.

Xu, X.; Zhang, Z.; Wang, Z.; Price, B. L.; Wang, Z.; and Shi, H. 2020. Rethinking Text Segmentation: A Novel Dataset and A Text-Specific Refinement Approach. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12040–12050.

Yang, M.; Yang, B.; Liao, M.; Zhu, Y.; and Bai, X. 2024. Class-Aware Mask-guided feature refinement for scene text recognition. *Pattern Recognition*, 149: 110244.

Ye, M.; Zhang, J.; Liu, J.; Liu, C.; Yin, B.; Liu, C.; Du, B.; and Tao, D. 2024. Hi-SAM: Marrying Segment Anything Model for Hierarchical Text Segmentation. arXiv:2401.17904.

Yu, D.; Li, X.; Zhang, C.; Liu, T.; Han, J.; Liu, J.; and Ding, E. 2020. Towards Accurate Scene Text Recognition With Semantic Reasoning Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12110–12119.

Yu, H.; Fu, T.; Li, B.; and Xue, X. 2024. EAFormer: Scene Text Segmentation with Edge-Aware Transformers. arXiv:2407.17020.

Zhang, W.; Deng, X.; Jia, B.; Yu, X.; Chen, Y.; jin Ma; Ding, Q.; and Zhang, X. 2023. Pixel Adapter: A Graph-Based Post-Processing Approach for Scene Text Image Super-Resolution. arXiv:2309.08919.

Zhao, M.; Wang, M.; Bai, F.; Li, B.; Wang, J.; and Zhou, S. 2022. C3-STISR: Scene Text Image Super-resolution with Triple Clues. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 1707–1713. Main Track.

Zhao, Z.; Xue, H.; Fang, P.; and Zhu, S. 2024. PEAN: A Diffusion-Based Prior-Enhanced Attention Network for Scene Text Image Super-Resolution. arXiv:2311.17955.

Zhou, Y.; Gao, L.; Tang, Z.; and Wei, B. 2023. Recognition-Guided Diffusion Model for Scene Text Image Super-Resolution. *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2940–2944.

Zhu, S.; Zhao, Z.; Fang, P.; and Xue, H. 2023a. Improving Scene Text Image Super-Resolution via Dual Prior Modulation Network. In *AAAI Conference on Artificial Intelligence*.

Zhu, X.; Guo, K.; Fang, H.; Ding, R.; Wu, Z.; and Schaefer, G. 2023b. Gradient-Based Graph Attention for Scene Text Image Super-resolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(3): 3861–3869.