PSTRFormer: Leveraging Segmentation and Attention for Improved Text Recognition in Complex Environments

XiaoMan Bai, and Ji Zhao*

Abstract—**Scene Text Recognition (STR) is a crucial task in computer vision, with applications spanning autonomous driving, intelligent surveillance, and document automation. Traditional Optical Character Recognition (OCR) struggles with scene text due to varying font styles, distortions, and complex backgrounds. We propose PSTRFormer (Progressive Scene Text Recognition Transformer), a novel STR framework that integrates edge-aware segmentation with deep learning-based recognition to address these challenges. The model comprises three primary modules: the Efficient and Accurate Scene Text (EAST) module, the SobelEdgeDetect module, and a multi-stage encoder-decoder network. The EAST module extracts preliminary segmentation features, while the SobelEdgeDetect module enhances text boundaries, improving localization accuracy. The encoder adopts a hierarchical multi-scale attention-based architecture, and the decoder reconstructs text segmentation masks with high fidelity. We evaluate PSTRFormer on benchmark datasets (ICDAR 2015, COCO-Text, and SynthText), achieving state-of-the-art performance with 96.47% Character Accuracy and 91.06% Word Accuracy on ICDAR 2015. Ablation studies confirm the efficacy of the edge-aware segmentation approach in enhancing text localization and recognition accuracy. Our findings demonstrate that integrating segmentation and recognition significantly improves STR performance, particularly in handling distorted, occluded, and curved text. This study paves the way for further advancements in real-world OCR applications.**

Index Terms—**Scene Text Recognition, Edge-aware Segmentation, Transformer-based Framework, Optical Character Recognition**

I. Introduction

SCENE Text Recognition (STR) is a computer vision task aimed at detecting and recognizing text from natural scene images[1–3]. Unlike traditional Optical Character Recognition (OCR), STR must handle challenges such as complex backgrounds, diverse font styles, blurriness, distortions, curvature, and occlusions, thereby necessitating higher generalization capability and robustness of models[4, 5]. This technology has extensive applications in autonomous driving, intelligent translation, smart surveillance, document automation, product recognition, and other fields.

Scene Text Recognition is divided into two primary steps: text detection and text recognition. Text detection aims to locate potential text regions within complex backgrounds, while text recognition converts the detected text regions into readable character sequences. These two tasks can be performed independently or jointly through end-to-end approaches to improve overall performance.

In text detection, traditional computer vision methods, such as Maximally Stable Extremal Regions (MSER) and Stroke Width Transform (SWT), perform well in simple scenarios but often fail to achieve satisfactory results in complex environments[6, 7]. Deep learning-based methods, particularly models utilizing Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have significantly improved the accuracy and robustness of text detection[8, 9]. Object detection and segmentation networks have been widely applied to text detection tasks in recent years. For example, EAST (Efficient and Accurate Scene Text Detector) directly predicts text box locations through regression, eliminating the need for additional post-processing steps[10–12]. CRAFT (Character Region Awareness for Text Detection) detects individual character regions and reconstructs words, achieving more precise detection. Furthermore, PSENet (Progressive Scale Expansion Network) and DBNet (Differentiable Binarization for Text Detection) have demonstrated superior performance in handling dense and extended text[13–15].

In text recognition, early approaches primarily relied on character-level recognition based on traditional OCR techniques. However, due to the deformations of characters and interference from complex backgrounds, these methods exhibited limited effectiveness in scene text recognition. With the advancement of deep learning, sequence-based learning methods have become mainstream. CRNN (Convolutional Recurrent Neural Network) integrates CNN and RNN architectures while employing Connectionist Temporal Classification (CTC) loss for training, effectively handling unaligned text sequences. Another category of methods is based on attention mechanisms in sequence-to-sequence (Seq2Seq) models, such as Attention OCR, which enables more precise recognition of irregular text. Recently, Transformer-based OCR methods, such as SATRN and PARSeq, have further improved text recognition performance. Leveraging self-attention mechanisms, these models capture global dependencies, making them particularly well-suited for recognizing deformed, blurred, or curved text.

Beyond standalone text detection and recognition models, end-to-end scene text recognition methods have also gained traction in recent years. Models such as Mask TextSpotter and ABINet jointly optimize detection and recognition tasks, mitigating error accumulation and enhancing overall recog-

Manuscript received Feb 17, 2025. revised May 18, 2025. The research work was supported by a scientific research project fund from the Liaoning Provincial Department of Education, and key project of Liaoning Provincial Department of Education(LJKZZ2022043)

XiaoMan Bai is a postgraduate student at the University of Science and Technology Liaoning, Anshan, Liaoning, China. (e-mail: BXM20001105@outlook.com).

Ji Zhao* is a Professor of University of Science and Technology Liaoning, Anshan, Liaoning, China. (corresponding author to provide phone: +086-139-9808-6167; e-mail: zhaoji_1974@126.com).

nition accuracy[16]. Additionally, VisionLAN (Visual Language Alignment Network) incorporates language modelling, improving the model's contextual understanding[17].

II. Related Work

A. SegFormer

SegFormer is an efficient semantic segmentation model that integrates the Transformer architecture with a lightweight multi-scale feature extraction mechanism, achieving a well-balanced trade-off between computational efficiency and accuracy[18]. Although SegFormer was initially designed for semantic segmentation tasks, its powerful feature extraction capability suggests potential applications in scene text recognition (STR).

In scene text recognition, the text is often embedded within complex backgrounds and may exhibit distortions and occlusions. Traditional CNN-based architectures can be constrained by their reliance on local features when processing such challenging conditions. In contrast, SegFormer, by leveraging the Transformer structure, can capture global contextual information, thereby enhancing the model's ability to comprehend text regions more effectively. Furthermore, its multi-scale feature extraction module enables the efficient detection of text regions of varying sizes and shapes, which is particularly advantageous for detecting curved text, densely arranged text, and small-scale text. For instance, in end-to-end text detection and recognition tasks, SegFormer can be employed for text region extraction, followed by final recognition using models such as CRNN or Transformer-based OCR.

Another significant advantage of SegFormer lies in its efficient architectural design. Compared to conventional Transformer-based structures such as ViT, SegFormer significantly reduces computational complexity, making it more suitable for real-time text detection tasks on mobile or embedded devices. By utilizing SegFormer for text region extraction in conjunction with other OCR recognition models, the robustness of scene text recognition can be further improved, particularly in scenarios involving text detection within complex backgrounds.

B. Sobel Operator

The Sobel operator is a widely used edge detection method that enhances edge information in an image by computing its gradient, thereby making text regions more distinguishable[19–21]. The fundamental principle of the Sobel operator involves applying convolutional kernels to calculate the image gradient in the horizontal direction (x-axis) and the vertical direction (y-axis), followed by integrating these gradients to extract edge information.

The horizontal gradient (Gx) and vertical gradient (Gy) of the Sobel operator are computed using the following Equations(1) and (2):

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \tag{1}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \tag{2}$$

During image processing, the Sobel operator is applied to an input image I(x,y) through convolution, yielding the horizontal gradient image Sx and the vertical gradient image Sy. This calculation process is shown in Equations (3) and (4).

$$S_x = I(x,y) * G_x S_y = I(x,y) * G_y \tag{3}$$

Where $*$ denotes the convolution operation. The final gradient magnitude is computed as Equations (5):

$$G = \sqrt{S_x^2 + S_y^2} \tag{4}$$

To reduce computational complexity, an approximate computation method can be employed by taking the sum of absolute values. This calculation process is shown in Equations (6).

$$G \approx |S_x| + |S_y| \tag{5}$$

Additionally, the gradient direction (θ) is given by Equations (7).

$$\theta = \tan^{-1}\left(\frac{S_y}{S_x}\right) \tag{6}$$

In the context of scene text recognition (STR), the Sobel operator is primarily used for text region preprocessing and feature enhancement to improve the accuracy of text detection and recognition. First, it enhances edge information in text regions, making them more prominent in complex backgrounds. In real-world scenarios, text is often embedded within intricate backgrounds, such as street signs, billboards, and natural scenes. The gradient images generated by the Sobel operator effectively emphasize text contours, facilitating more accurate text detection by models such as EAST, CRAFT, and DBNet, particularly in low-contrast or noisy environments.

Furthermore, the Sobel operator can be incorporated as an auxiliary input in deep learning-based OCR models to enhance text recognition performance. Traditional OCR pipelines typically process raw RGB images, whereas modern OCR architectures, such as CRNN and Transformer-based OCR, can integrate the gradient information derived from the Sobel operator as an additional input channel. This approach enables the model to capture richer structural information of text, which is particularly beneficial in scenarios involving blurred character edges or significant illumination variations, thereby improving model generalization.

Beyond direct applications in text edge enhancement and OCR feature augmentation, the Sobel operator can also contribute to segmentation-based text detection. When segmentation-based models such as SegFormer or other deep learning-based text detection frameworks are used, Sobel edge detection can be applied to refine text region boundaries in conjunction with segmentation results. This technique reduces false positives and enhances the localization accuracy of irregular text regions, making it especially effective for detecting curved and densely arranged text.

The Sobel operator is a valuable enhancement tool in text detection and recognition. By providing additional gradient information, it can be employed in traditional OCR pipelines for text edge enhancement or integrated with deep learning

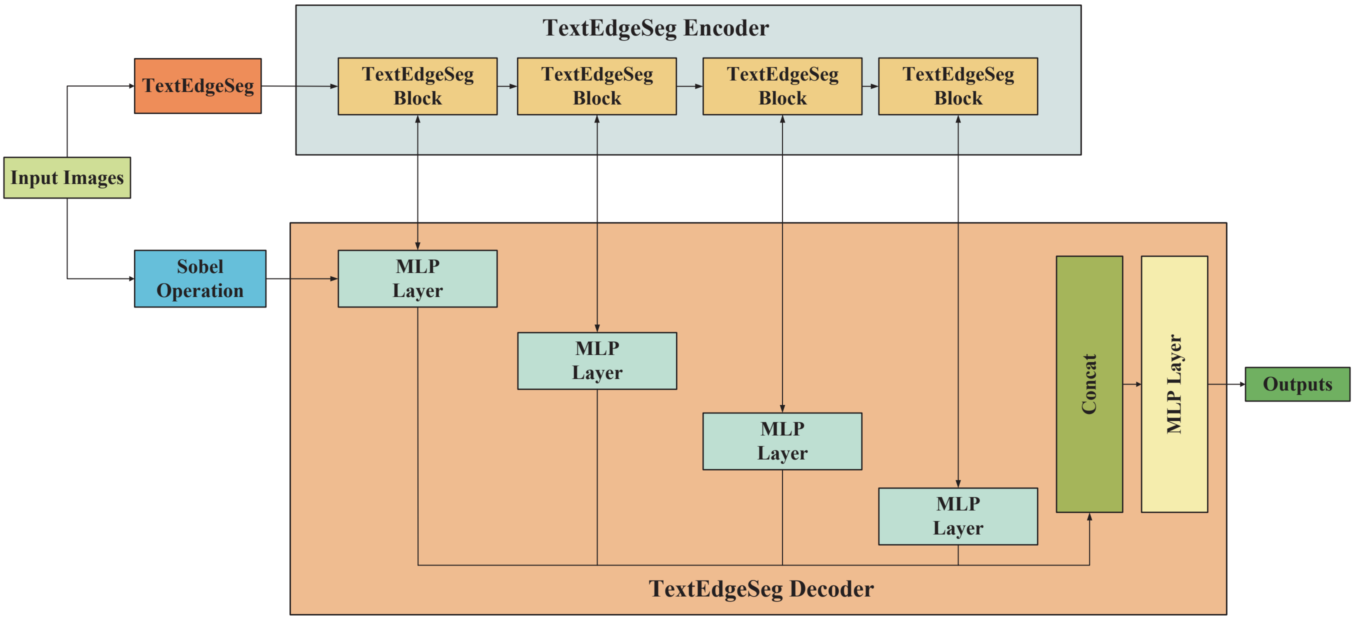

Fig. 1. The Structure of PSTRFormer

models as a complementary feature. The Sobel operator significantly contributes to the robustness and accuracy of scene text recognition systems in complex environments.

C. Text recognition technology based on attention mechanism

In scene text recognition, the attention mechanism plays a pivotal role, particularly in recognising distorted, occluded, and long text sequences. Traditional OCR methods predominantly rely on CNNs and RNNs for feature extraction and sequence modelling. However, these approaches often exhibit limited effectiveness when confronted with irregular text in complex scenes—such as rotated or curved text. The introduction of the attention mechanism enables models to dynamically focus on different regions of the text, thereby improving recognition accuracy.

Attention-based text recognition methods are typically integrated with the Seq2Seq architecture, particularly in RNN-based Attention OCR and Transformer-based OCR models. In the RNN + Attention structure, a CNN first extracts the visual features of the text, followed by sequence modelling via an RNN. The attention mechanism is then applied during decoding, allowing the model to focus on different spatial regions of the image at each time step and output the corresponding character. This approach effectively aligns the input text image with the output sequence, enhancing the recognition capability for distorted and irregular text.

In contrast, Transformer-based OCR models (e.g., SATRN, PARSeq) leverage the self-attention mechanism to simultaneously capture both global and local features of the text image, eliminating the need for RNN-based sequence modelling and reducing computational complexity in training and inference. These methods employ a Transformer encoder to extract text features, followed by an attention-based decoder to predict the character sequence. This approach is well-suited for recognizing long text sequences and complex text layouts. Additionally, some end-to-end OCR models (e.g., ABINet, VisionLAN) integrate visual attention and language modelling, enabling the recognition system to rely on image features and leverage linguistic context to refine recognition results. These models can correct misrecognized characters by predicting probable word structures, further improving recognition accuracy.

In practical applications, the attention mechanism extends beyond text recognition and is employed in text detection and rectification. For instance, in text detection models based on CRAFT or SegFormer, the attention mechanism enhances the localization of text regions, particularly in scenarios involving dense text clusters and small-scale text. Furthermore, in-text rectification tasks and attention mechanisms assist in adjusting the geometric structure of the text, ensuring a more standardized representation and improving final recognition accuracy.

The attention mechanism has become a fundamental technology in scene text recognition, significantly enhancing model adaptability in complex textual environments. Its advantages are particularly evident in recognizing distorted, long, and multilingual text. As Transformer architectures continue to be refined, future OCR recognition systems are expected to become more efficient and accurate, playing an increasingly critical role in real-world applications.

III. PSTRFormer

A. Overall Process

This study proposes a progressive scene text recognition framework based on edge-aware segmentation, termed PSTRFormer (Progressive Scene Text Recognition Transformer), designed to address the challenges of recognizing text in complex scene environments. Figure 1 illustrates the structural diagram of the framework. A unified segmentation framework is constructed by integrating edge information and segmentation features. The overall architecture of PSTRFormer consists of three primary modules: the Efficient and Accurate Scene Text (EAST) module, the SobelEdgeDetect module, and a multi-stage encoder-decoder network.

The Efficient and Accurate Scene Text module primarily extracts preliminary segmentation features and generates coarse predictions of text regions, laying the foundation for

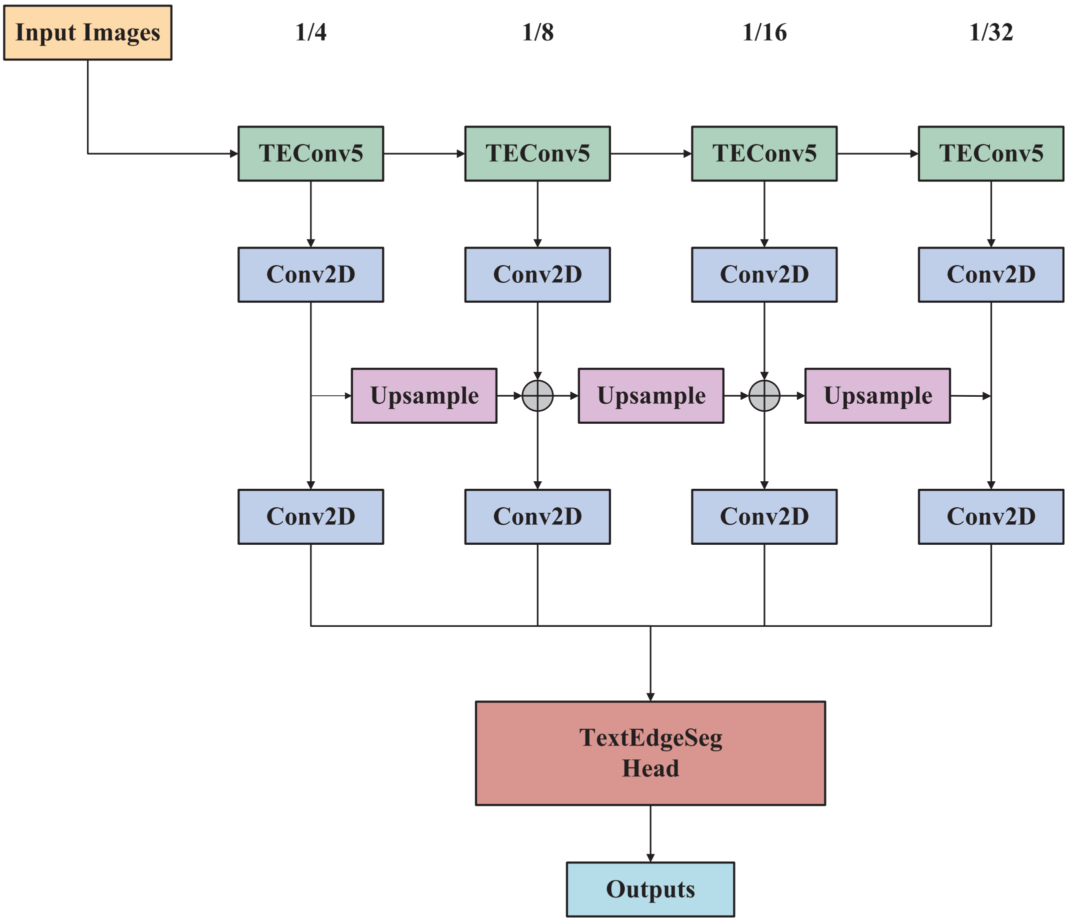

Fig. 2. The Structure of TextEdgeSeg

subsequent fine-grained segmentation. Simultaneously, the SobelEdgeDetect module employs Sobel edge detection to capture text boundary information, enhancing the model's ability to delineate text contours accurately. These two components complement each other, providing diverse sources of information to facilitate high-precision text segmentation.

The encoder is designed as a hierarchical stack of text segmentation blocks (Efficient and Accurate Scene Text Blocks) for feature extraction. At the same time, the decoder adopts a multi-path upsampling architecture to reconstruct high-quality text segmentation masks. Finally, the output layer, implemented as an MLP, generates consistent-resolution segmentation results with well-defined text boundaries.

B. *Efficient and Accurate Scene Text Module*

The Efficient and Accurate Scene Text module is a key component within the PSTRFormer framework, and it is responsible for extracting text edge information and generating initial text segmentation predictions. The structure of Text Edge Segment is shown in Figure 2. This module employs a multi-scale feature extraction and fusion strategy to capture text boundary details accurately. The input image is processed through multiple feature extraction pathways at different resolutions (1/4, 1/8, 1/16, and 1/32 of the original resolution). Each pathway incorporates TEConv5 modules designed to capture rich contextual information. The multi-resolution processing ensures the model's multi-level perception of text boundaries, particularly excelling in scenarios involving complex artistic font structures. Figure 3 shows the structure of TEConv5.

During the feature fusion stage, the Efficient and Accurate Scene Text module employs a progressive upsampling and additive fusion strategy, where low-resolution features are gradually restored to higher-resolution spatial scales (e.g., ×2, ×4, ×8), enabling efficient integration of information across different resolutions. The final multi-scale fused features undergo a series of convolutional operations, culminating in the Text Edge Segment Head, which generates a binarized preliminary text segmentation mask to localize text regions coarsely.

With its multi-scale processing capability and lightweight design, this module demonstrates superior edge localization and feature representation performance, providing a robust foundation for subsequent fine-grained text segmentation.

C. *Text Segmentation Encoder*

The Text Segmentation Encoder is the core component of the PSTRFormer framework. It is responsible for extracting deep semantic features from the input image and generating multi-scale feature maps to provide robust feature representations for fine-grained text segmentation.

As illustrated in Figure 4, each Efficient and Accurate Scene Text Block consists of two key modules: Efficient Self-Attn and Mix-FFN, which dynamically model features across spatial and channel dimensions. This module is inspired by the SegFormer Block, leveraging its powerful contextual modelling capabilities and efficient multi-scale feature extraction mechanism to handle complex text structures effectively.

The Text Segmentation Encoder progressively extracts features at different resolutions, capturing local text details and global contextual information by stacking multiple Efficient and Accurate Scene Text Blocks. Additionally, the encoder integrates edge information from the SobelEdgeDetect module and the Efficient and Accurate Scene Text module, enhancing the boundary sensitivity of the generated feature maps. The final output of the encoder consists of multi-scale features, providing high-quality input to the decoder, thereby ensuring both semantic consistency and boundary precision in the segmentation process.

D. Text Segmentation Decoder

The Text Segmentation Decoder is designed as a multi-path upsampling architecture to reconstruct high-quality text segmentation masks. Initially, an MLP layer processes each scale of the extracted features. Subsequently, a progressive upsampling pathway (×2, ×4, ×8) is employed to restore the feature maps' spatial resolution gradually. Once the multi-scale features are aligned, a feature fusion module integrates information across different scales. Finally, the output MLP layer generates a consistent-resolution text segmentation result with well-defined boundaries.

IV. Experiment Settings

A. Datasets

1) ICDAR 2015 Dataset: The ICDAR 2015 dataset (ICDAR Incidental Scene Text) is a scene text recognition dataset introduced by the International Conference on Document Analysis and Recognition (ICDAR), focusing on the detection and recognition of unstructured text. Unlike its predecessor, ICDAR 2013, ICDAR 2015 primarily comprises randomly captured text images sourced from handheld camera recordings of street scenes, billboards, and storefront signage. These images present significant challenges, including motion blur, occlusion, skew, and deformation.

The dataset contains 1,500 images (1,000 for training and 500 for testing), all precisely annotated with word-level quadrilateral bounding boxes and complete transcription. A distinguishing characteristic of ICDAR 2015 is its lack of structured text alignment, with text appearing in various orientations and deformations, posing significant challenges for traditional OCR methods. Consequently, it has become a critical benchmark for evaluating modern deep-learning-based OCR models.

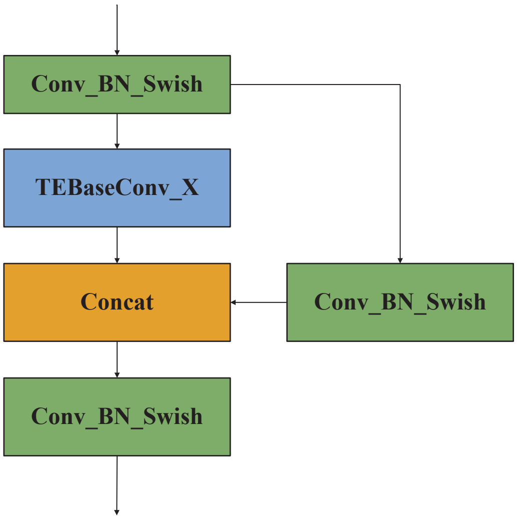

Fig. 3. The Structure of Conv5

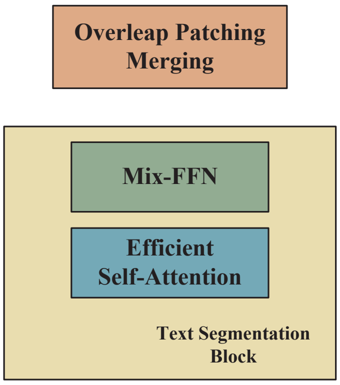

Fig. 4. The Structure of Text Segmentation

2) COCO-Text Dataset: The COCO-Text dataset extends the COCO (Common Objects in Context) dataset, which was designed explicitly for text detection and recognition in natural scenes. It comprises 63,686 images containing 239,506 text instances, categorised as legible, illegible, or no text.

A key feature of COCO-Text is its highly complex scene variability. Text appears in diverse contexts such as signboards, advertisements, product packaging, T-shirts, and walls, exhibiting various fonts, colours, sizes, and languages. Additionally, COCO-Text provides categorical annotations, distinguishing between handwritten and machine-printed text, making it a comprehensive dataset for OCR research.

Due to its diversity and complexity, COCO-Text is valuable for robustness evaluation and multi-context text detection. It is an essential dataset for training and assessing adaptive text recognition models.

3) SynthText Dataset: The SynthText dataset is a large-scale synthetic dataset designed for training deep learning models to enhance generalization performance in scene text recognition tasks. It comprises over 800,000 synthetically generated images, with simulated text embedded into real-world backgrounds.

SynthText generates images by randomly overlaying text on authentic images while applying varied fonts, colours, backgrounds, and distortions, ensuring text appearance closely resembles real-world scenarios. This approach significantly expands the available OCR training data volume, thereby enhancing model stability and generalization on real datasets.

Furthermore, SynthText provides precise character-level, word-level, and text-region annotations, making it suitable for various OCR tasks, including text detection, recognition, and end-to-end OCR training. Due to the scalability of synthetic data, SynthText is widely used for pretraining deep-learning-based OCR models such as CRNN, Transformer OCR, EAST, and CRAFT, ultimately improving performance on real-world datasets.

B. Merits

In scene text recognition (STR) tasks, commonly used evaluation metrics include Accuracy, Edit Distance, Recall, Precision, and F1 Score. These metrics assess a model's performance in text detection and recognition, ensuring its robustness and accuracy in complex scene settings.

1) Accuracy: Accuracy is one of the most widely adopted evaluation criteria and can be computed at both the character level (Character Accuracy, CA) and the word level (Word Accuracy, WA). Character Accuracy measures the model's ability to recognize individual characters, whereas Word Accuracy requires the entire word to be correctly identified. In OCR tasks, word-level accuracy (WA) is often considered more representative since the ultimate goal is to accurately recognize entire phrases or sentences rather than merely individual characters. The formulas for CA and WA are presented in Equations (2) and (3).

$$CA = \frac{\text{Number of correctly recognized characters}}{\text{Total number of characters}} \times 100\% \tag{7}$$

$$WA = \frac{\text{Number of correctly recognized words}}{\text{Total number of words}} \times 100\% \tag{8}$$

Word-level accuracy imposes stricter requirements than character-level accuracy, as even a single incorrect character in a word results in recognition failure for that word. This metric is more intuitive in practical applications, as users prioritize the recognition accuracy of entire words over individual character matching.

2) Edit Distance: Edit Distance is a crucial metric for evaluating the similarity between the recognition result and the ground-truth text. It is commonly computed using the Levenshtein Distance, which quantifies the minimum number of edit operations required to transform the recognized text into the correct text. The allowed operations include Insertion (adding a character), Deletion (removing a character) and Substitution (replacing a character)

A lower edit distance indicates a higher similarity between the recognition result and the ground truth. The edit distance calculation is expressed in Equation (4):

$$\text{Edit Distance=Min(insertions, deletions,substitutions)} \tag{9}$$

To enable comparisons across texts of varying lengths, a Normalized Edit Distance (NED) is often used, calculated as Equation (5):

$$NED = 1 - \frac{\text{Levenshtein Distance}}{\text{Length of the ground truth text}} \tag{10}$$

The NED value ranges from 0 to 1, where 1 signifies a perfect match, and 0 indicates an incorrect recognition result.

Edit Distance's advantage is its ability to quantify OCR errors precisely rather than rely solely on binary correctness checks. This metric provides a fine-grained evaluation in long text recognition tasks, making it a widely used complement to accuracy-based assessments in OCR research.

3) Precision and Recall: Precision evaluates the proportion of correctly detected text regions among all detected areas. It is computed as follows Equation (6):

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

Where TP (True Positives) means Correctly detected text regions. FP (False Positives) means Incorrectly detected text regions.

A higher precision indicates a lower false detection rate, signifying that the model makes fewer incorrect predictions.

Recall measures the proportion of actual text regions that are correctly detected. It is computed as Equation (7):

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

Where FN (False Negatives) means Undetected text regions

A higher recall suggests that the model detects a significant portion of text regions, but an excessively high recall may lead to more false positives.

F1 Score

The F1 Score is the harmonic mean of Precision and Recall, providing a comprehensive assessment of detection performance. It is computed as Equation (8):

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

A higher F1 Score indicates a more balanced trade-off between Precision and Recall, reflecting strong overall performance in text detection tasks.

4) FPS (Frames Per Second): FPS measures the number of images the model can process per second, indicating inference speed. It is computed as Equation (9):

$$FPS = \frac{\text{Number of processed frames}}{\text{Time taken (in seconds)}} \tag{14}$$

A higher FPS signifies a faster model crucial for real-time OCR applications.

5) FLOPs (Floating Point Operations per Second): FLOPs represent the number of floating-point operations required for model inference, measuring its computational complexity. Unlike FPS, FLOPs do not indicate execution speed but instead reflect the computational burden of the model. It is computed as Equation (10):

$$FLOPs = \sum_{i=1}^{N} (\text{Operations per layer} \times \text{Number of layers}) \tag{15}$$

FLOP values are typically reported in MFLOPs (Million FLOPs) to GFLOPs (Billion FLOPs), where a higher FLOP value signifies greater computational demand but potentially higher accuracy.

The range and magnitude of each evaluation metric directly impact the performance and effectiveness of the model. Accuracy (CA and WA) ranges from 0% to 100%, where higher values indicate better recognition performance, with WA being particularly representative. Edit Distance should be as low as possible, with 0 indicating a perfect match, while the Normalized Edit Distance (NED) ranges from 0 to 1, where values closer to 1 indicate higher recognition accuracy. Precision and Recall range from 0 to 1, with higher Precision indicating fewer false positives and higher Recall indicating fewer missed detections. A higher F1 Score represents better overall text detection performance. FPS (Frames Per Second) should be higher to ensure faster inference speed, making the model suitable for real-time OCR applications. FLOPs (Floating Point Operations per Second) are typically measured in MFLOPs to GFLOPs, where a higher value indicates greater computational complexity, potentially leading to improved accuracy but increased computational demands. Ideally, the model should achieve CA/WA close to 100%, low Edit Distance (NED close to 1), high F1 Score, high FPS, and moderate FLOPs to balance accuracy and efficiency. To ensure consistency in data representation, all evaluation metrics except FPS and FLOPs are expressed as percentages.

C. Baseline

1) CRNN (Convolutional Recurrent Neural Network) is a text recognition model that integrates Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), initially proposed by Shi et al. CRNN first employs a CNN to extract visual features from images, followed by bidirectional LSTM for sequence modelling. Then, it is trained using Connectionist Temporal Classification (CTC) loss. This approach handles text without fixed alignment, making it suitable for scene text recognition in naturally arranged words. However, CRNN relies on RNNs for sequence modelling, resulting in slower inference speed and difficulty processing curved or distorted text.

2) SATRN (Self-Attention Text Recognition Network) represents a mainstream advancement in Scene Text Recognition (STR) tasks. It eliminates the RNN component, relying entirely on a Transformer-based architecture. Using self-attention mechanisms, SATRN learns global text features, significantly enhancing parallel computing capability and inference speed. SATRN excels in handling long, deformed, and text in complex backgrounds, making it well-suited for high-precision OCR tasks. However, due to the high computational complexity of the Transformer architecture, SATRN has a large FLOP requirement, posing challenges for deployment on edge devices.

3) Mask TextSpotter is an end-to-end text detection and recognition framework based on Mask R-CNN. It integrates text instance segmentation and sequence recognition to handle curved and complexly arranged text in scenes effectively. It employs a multi-task learning architecture, jointly optimizing text detection (localization) and text recognition, thereby mitigating error accumulation between the detection and recognition modules. This design is particularly effective for irregular text, including curved and rotated text, making it a robust solution for complex OCR scenarios.

V. Result and Analysis

A. Model Comparison

First, this study conducted comparative experiments between the baseline models and PSTRFormer on the previously mentioned datasets, with the results presented in Table 1. The PSTRFormer model demonstrated significant superiority over other models on three benchmark datasets: ICDAR 2015, COCO-Text, and SynthText.To present the experimental results more intuitively, we employed data visualization techniques such as Figure 5 to enhance the clarity and impact of our findings. Using the ICDAR 2015 dataset as an example, we normalized the performance metrics of PSTRFormer to 100 as a baseline and compared the relative performance of other models, including CRNN, SATRN, and Mask TextSpotter, across key indicators such as Character Accuracy, Word Accuracy, Normalized Edit Distance, Precision, Recall, and F1 Score. This visualization approach effectively highlights the performance gaps between models, improving the readability and persuasive power of the experimental analysis.

On the ICDAR 2015 dataset, PSTRFormer achieved a Character Accuracy (CA) of 96.47%, surpassing CRNN (92.15%), SATRN (93.44%), and Mask TextSpotter (93%) by 4.32%, 3.03%, and 3.47%, respectively. The Word Accuracy (WA) reached 91.06%, representing a 6.73% improvement over CRNN (84.33%) and exceeding SATRN (89.35%) and Mask TextSpotter (88.14%). Additionally, Normalized Edit Distance (NED) reached 97.43%, the highest among all models, highlighting PSTRFormer's exceptional ability in text edge information extraction. Precision (92.65%) and Recall (89.78%) also ranked highest, while its F1 score (93.65%) further demonstrated its outstanding overall performance.

On the COCO-Text dataset, PSTRFormer continued to lead, achieving a CA of 98.01%, outperforming CRNN (93.49%) and other models. The WA and NED scores, reaching 97.29% and 99.48%, respectively, underscored its robustness and efficiency in more complex scenarios. The F1 score of 97.03% significantly exceeded those of competing models, fully demonstrating its capabilities in diverse text detection tasks.

On the SynthText synthetic dataset, PSTRFormer also exhibited remarkable performance, with CA and WA reaching 96.74% and 96.76%, respectively. Precision (94.63%) and F1 score (95.93%) further consolidated its leading position in multi-scene and diversified OCR tasks.

The superior performance of PSTRFormer stems from its innovative module design and efficient feature processing mechanism. The Efficient and Accurate Scene Text module employs multi-resolution feature extraction pathways and the TEConv5 module to achieve multi-scale precise text edge detection. Through a progressive upsampling and additive fusion strategy, this module effectively integrates multi-resolution features, ensuring high-precision text edge localization, particularly excelling in complex and artistic

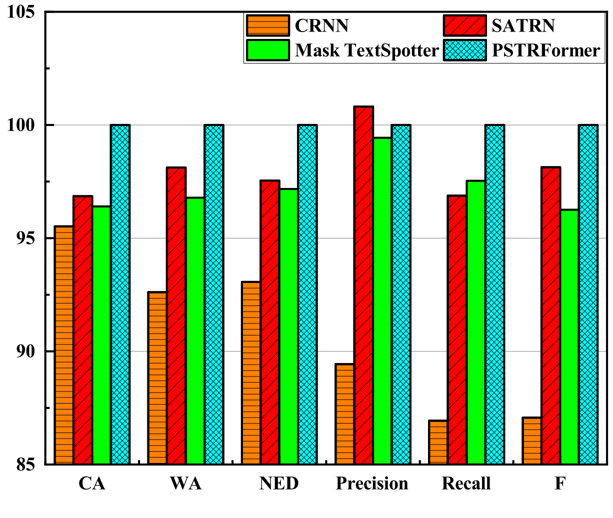

Fig. 5. Proportional Performance Comparison of Baseline Models

TABLE I
SUPERIORITY OF PSTRFORMER IN MULTI-DATASET SCENE TEXT RECOGNITION BENCHMARKS

Datasets	Merits	Models			
		CRNN	SATRN	Mask TextSpotter	PSTRFormer
ICDAR 2015	CA	92.15	93.44	93	**96.47**
	WA	84.33	89.35	88.14	**91.06**
	NED	90.67	95.03	94.67	**97.43**
	Precision	82.87	**93.4**	92.13	92.65
	Recall	78.05	86.98	87.56	**89.78**
	F1	81.54	91.9	90.15	**93.65**
	FLOPs	**10.34**	27.35	34.57	33.76
	FPS	**37.5**	19.2	17.6	20
COCO-Text	CA	93.49	92.29	93.85	**98.01**
	WA	83.27	90.07	88.57	**93.69**
	NED	88.88	95.65	93.25	**99.48**
	Precision	84.96	91.27	93.12	**97.29**
	Recall	77.98	87.05	89.28	**94.83**
	F1	83.90	91.17	91.67	**97.03**
	FLOPs	**13.56**	27.21	35.50	37.49
	FPS	**39.30**	18.50	17.60	23.40
SynthText	CA	89.67	91.03	91.62	**95.31**
	WA	84.27	**91.70**	86.04	89.76
	NED	92.58	95.68	93.78	**96.74**
	Precision	83.44	91.74	94.26	**98.56**
	Recall	80.37	86.50	88.92	**94.63**
	F1	79.49	92.60	92.10	**95.93**
	FLOPs	**12.77**	26.79	35.24	36.31
	FPS	**33.50**	19.10	17.50	23.40

TABLE II
ABLATION RESULTS OF PSTRFORMER'S KEY MECHANISMS ACROSS MULTIPLE METRICS

			Merits							
	TextEdgeSeg	SobelEdgeDetect	CA	WA	NED	Precision	Recall	F1	FLOPs	FPS
Group 1	×	×	91.33	85.47	89.15	87.08	86.74	89.53	**29.86**	**22**
Group 2	✓	×	94.48	88.02	92.93	90.13	86.34	90.72	31.34	22
Group 3	×	✓	92.56	86.27	90.42	91.49	88.73	92.35	31.87	21
Group 4	✓	✓	**96.38**	**91.11**	**97.42**	**92.61**	**89.78**	**93.62**	33.76	20

font structures. By stacking Efficient Self-Attn and Mix-FFN modules, the Text Segmentation Encoder dynamically models features across spatial and channel dimensions, enhancing semantic consistency and boundary sensitivity. The SobelEdgeDetect module and edge information fusion further improve edge feature accuracy, laying a solid foundation for text segmentation in complex environments. The Text Segmentation Decoder, utilizing a multi-path upsampling architecture and a feature fusion module, progressively restores high-resolution features, generating consistent-resolution text segmentation results with sharp and well-defined boundaries.

Despite PSTRFormer's significant performance advantages, it incurs a slightly higher computational cost than lightweight models such as CRNN in specific scenarios. Additionally, its generalization capability for unseen data distributions (e.g., new fonts or specialized languages) requires further validation. Furthermore, under extreme conditions (e.g., low-resolution images or severe noise), there remains room for improvement in model robustness.

In summary, PSTRFormer achieves a technological breakthrough in scene text detection through multi-scale feature extraction, fusion, and enhanced boundary sensitivity. Its notable improvements in accuracy, edge detection precision, and overall performance establish it as a significant advancement in the field.

B. Ablation Experiment

Following Experiment 1, a series of ablation experiments were conducted on the ICDAR 2015 dataset to validate the contributions of the two core modules in the PSTRFormer model. The results, presented in Table 2, demonstrate the critical role of TextEdge and SobelEdgeDetect in improving the model's overall performance. By analyzing four experimental groups, the independent contributions of each module, as well as their synergistic effects, can be clearly understood.

In Group 1, where neither TextEdge nor SobelEdgeDetect was enabled, the model achieved a Character Accuracy (CA) of 91.33%, a Word Accuracy (WA) of 85.47%, and a Normalized Edit Distance (NED) of 89.15%. Although the model was able to perform essential text detection, its Precision (87.08%), Recall (86.74%), and F1 score (86.74%) remained at a fundamental level. The FLOP was 29.86, and the FPS reached 22, indicating a relatively low computational cost but suboptimal performance.

The model's performance improved significantly in Group 2, where only the TextEdge module was enabled. The CA increased from 91.33% to 94.48%, and the WA rose from 85.47% to 88.02%, demonstrating the TextEdge module's effectiveness in enhancing text classification and weighted accuracy. Additionally, the NED improved to 92.93%. In comparison, Precision (90.13%), Recall (86.34%), and F1 score (90.72%) all increased, indicating that the TextEdge module, through its multi-scale feature extraction strategy, effectively enhanced the capture of text edge information. The FLOPs slightly increased to 31.34, while the FPS remained at 22, suggesting minimal computational overhead increase.

In Group 3, where only the SobelEdgeDetect module was enabled, the model's CA further increased to 92.56%, while WA reached 86.27%. Compared to Group 2, this configuration exhibited slightly higher performance in NED (90.42%) and Recall (88.73%), with Precision reaching 91.49% and an F1 score of 92.35%. These results indicate that the SobelEdgeDetect module improved the model's boundary sensitivity and contextual feature modelling capability by precisely extracting edge information. Regarding computational cost, FLOPs reached 31.87, and FPS decreased slightly to 21, reflecting a marginal increase in complexity over Group 2.

The model achieved the best performance in Group 4, where both TextEdge and SobelEdgeDetect modules were enabled. The CA significantly increased to 96.38%, WA reached 91.11%, and NED achieved 97.42%, the highest among all experimental groups. Additionally, the Precision (92.61%), Recall (89.78%), and F1 score (93.62%) demonstrated the advantages of the synergistic effect of both modules. The FLOPs increased slightly to 33.76, while the FPS decreased to 20, indicating that the performance gains came at the expense of higher computational complexity. However, this trade-off was justified, as the substantial improvement in performance holds significant implications for scene text detection in complex environments.

In conclusion, the ablation study confirms that the TextEdge module is key in multi-scale feature extraction and edge information enhancement. In contrast, the SobelEdgeDetect module further improves boundary recognition through enhanced edge sensitivity and fine-grained feature modelling. When these two modules operate together, the model fully exploits its potential, achieving optimal results across multiple key metrics, particularly in edge detection and segmentation tasks in challenging text scenarios.

C. Hyperparameter Settings

Following the ablation experiments, a series of hyperparameter selection experiments were conducted on the ICDAR 2015 dataset to evaluate the impact of the hyperparameter λ on the performance of the PSTRFormer model. The results are presented in Table 3, demonstrating the significant influence of λ on scene text recognition tasks. The following is a systematic analysis based on the experimental data:

CA and WA are key performance metrics for evaluating the effectiveness of scene text recognition models, measuring recognition capability at the character level and word level, respectively. The experimental results show a clear trend as changes: when λ = 1.2, CA reaches its optimal value of 96.45%, and when λ = 1.0, WA reaches its optimal value of 92.45%.This finding suggests that an appropriate hyperparameter setting enables the model to balance character-level and word-level recognition tasks optimally.

NED is an important metric for quantifying the similarity between the predicted and ground-truth sequences. At λ=1.2, NED reaches 97.37%, the highest among all experimental settings. This indicates that the model generates prediction sequences closer to the ground truth under this hyperparameter configuration, significantly reducing the edit distance and validating its effectiveness in scene text recognition tasks.

In terms of Precision and Recall, the model's performance fluctuates with changes in λ=1.2; Precision and Recall achieve their highest values at 92.65% and 89.80%, respectively, indicating a well-balanced trade-off between reducing false positives and controlling false negatives. Moreover, the F1 score also reaches its peak at 93.65%, further confirming that this hyperparameter setting optimizes the model's overall performance.

Additionally, analysis of FLOPs and FPS reveals that these two computational efficiency metrics remain stable regardless of the variation in $lambda$. FLOPs consistently remain around 33.7G, while FPS varies between 19 and 20, indicating that adjusting $lambda$ minimally impacts computational resource requirements and operational efficiency. This ensures the model maintains high recognition performance while preserving practical feasibility for real-world applications.

The PSTRFormer model exhibits notable sensitivity to the hyperparameter $lambda$ in scene text recognition tasks. When λ=1.2, the model achieves optimal performance across character-level, word-level, and sequence-level metrics, suggesting that this hyperparameter value enables optimal recognition effectiveness while ensuring controlled computational resource consumption. These experimental findings provide valuable insights for the practical deployment of the model.

VI. Conclusions

In this study, we propose PSTRFormer, a progressive scene text recognition framework designed to improve recognition accuracy in complex environments through edge-aware segmentation and deep learning-based text recognition. The model integrates three key modules: the EAST text detection module, the SobelEdgeDetect module, and a multi-stage encoder-decoder network. Our experiments demonstrate that leveraging edge information alongside segmentation-based recognition enhances overall performance by refining text

TABLE III
HYPERPARAMETER COMPARISON

λ	Merits							
	CA	WA	NED	Precision	Recall	F1	FLOPs	FPS
0.1	93.46	85.87	90.04	89.53	83.15	88.51	**33.69**	19.00
0.5	94.14	88.56	92.85	91.72	84.91	90.87	33.71	**20.00**
1.0	95.47	**92.45**	96.14	91.33	86.54	**93.84**	33.70	20.00
1.2	**96.45**	91.10	**97.37**	**92.65**	**89.80**	93.65	33.73	20.00
5.0	93.65	87.03	94.62	90.08	88.56	91.57	33.72	19.00
10.0	89.33	82.34	91.03	87.02	84.36	86.53	33.70	20.00

boundary localization and improving feature extraction in challenging scenarios.

We evaluate PSTRFormer on three benchmark datasets—ICDAR 2015, COCO-Text, and SynthText—where it outperforms traditional OCR models such as CRNN, SATRN, and Mask TextSpotter. On ICDAR 2015, PSTRFormer achieves a 96.47% Character Accuracy and a 97.43% Normalized Edit Distance, significantly surpassing previous models. The ablation study further confirms the contributions of the TextEdge and SobelEdgeDetect modules, with their combined effect yielding the highest accuracy improvements.

Despite its superior accuracy, PSTRFormer introduces a slight increase in computational complexity compared to lightweight models like CRNN. However, its substantial performance gains justify the trade-off, particularly in recognizing irregular, blurred, and curved text. Additionally, the hyperparameter analysis reveals that an optimized parameter setting ($lambda$ = 1.2) maximizes accuracy while maintaining computational efficiency.

Future work will optimize model efficiency for real-time applications, improve robustness to unseen fonts and languages, and integrate self-supervised learning techniques to reduce reliance on labelled data. This research demonstrates that combining segmentation, attention mechanisms, and edge detection offers a powerful approach to scene text recognition, paving the way for more robust and accurate OCR solutions in diverse real-world applications.

REFERENCES

[1] A. Aberdam, D. Bensaïd, A. Golts, R. Ganz, O. Nuriel, R. Tichauer, S. Mazor, and R. Litman, “Clipter: Looking at the bigger picture in scene text recognition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 21 706–21 717.

[2] N. Nguyen, T. Nguyen, V. Tran, M.-T. Tran, T. D. Ngo, T. H. Nguyen, and M. Hoai, “Dictionary-guided scene text recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7383–7392.

[3] Y. He, C. Chen, J. Zhang, J. Liu, F. He, C. Wang, and B. Du, “Visual semantics allow for textual reasoning better in scene text recognition,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 888–896.

[4] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, “Trocr: Transformer-based optical character recognition with pre-trained models,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 11, 2023, pp. 13 094–13 102.

[5] S. Srivastava, A. Verma, and S. Sharma, “Optical character recognition techniques: A review,” in *2022 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS)*. IEEE, 2022, pp. 1–6.

[6] K. Dutta, R. Sarkhel, M. Kundu, M. Nasipuri, and N. Das, “Natural scene text localization and detection using mser and its variants: a comprehensive survey,” *Multimedia Tools and Applications*, vol. 83, no. 18, pp. 55 773–55 810, 2024.

[7] T. Tyagi, P. Gupta, and P. Singh, “A hybrid multi-focus image fusion technique using swt and pca,” in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020, pp. 491–497.

[8] N. Ketkar, J. Moolayil, N. Ketkar, and J. Moolayil, “Convolutional neural networks,” *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch*, pp. 197–242, 2021.

[9] I. D. Mienye, T. G. Swart, and G. Obaido, “Recurrent neural networks: A comprehensive review of architectures, variants, and applications,” *Information*, vol. 15, no. 9, p. 517, 2024.

[10] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, “East: an efficient and accurate scene text detector,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[11] T. Sheng, J. Chen, and Z. Lian, “Centripetaltext: An efficient text instance representation for scene text detection,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 335–346, 2021.

[12] Y. Su, “Efficient and accurate scene text detection with low-rank approximation network,” *ArXiv Preprint ArXiv:2306.15142*, 2023.

[13] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, “Character region awareness for text detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9365–9374.

[14] H. Nguyen, D. Tran, K. Nguyen, and R. Nguyen, “Psenet: Progressive self-enhancement network for unsupervised extreme-light image enhancement,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 1756–1765.

[15] Q. Mu and S. Wang, “Gradient adjustment for better differentiable binarization in scene text detection,” in *2024 5th International Conference on Artificial Intelligence and Electromechanical Automation (AIEA)*. IEEE, 2024, pp. 785–789.

[16] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, “Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 67–83.

[17] Y. Gao, J. Liu, Z. Xu, J. Zhang, K. Li, R. Ji, and C. Shen, “Pyramidclip: Hierarchical feature alignment for vision-language model pretraining,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 959–35 970, 2022.

[18] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[19] L. Han, Y. Tian, and Q. Qi, “Research on edge detection algorithm based on improved sobel operator,” in *MATEC Web of Conferences*, vol. 309. EDP Sciences, 2020, p. 03031.

[20] G. Ravivarma, K. Gavaskar, D. Malathi, K. Asha, B. Ashok, and S. Aarthi, “Implementation of sobel operator based image edge detection on fpga,” *Materials Today: Proceedings*, vol. 45, pp. 2401–2407, 2021.

[21] S. Hao, B. Wu, K. Zhao, Y. Ye, and W. Wang, “Two-stream swin transformer with differentiable sobel operator for remote sensing image classification,” *Remote Sensing*, vol. 14, no. 6, p. 1507, 2022.