IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 February 2024, accepted 5 March 2024, date of publication 11 March 2024, date of current version 21 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3375497

RESEARCH ARTICLE

Transforming Scene Text Detection and Recognition: A Multi-Scale End-to-End Approach With Transformer Framework

TIANYU GENG
College of Computer and Information Engineering (College of Artificial Intelligence), Nanjing Tech University, Nanjing, Jiangsu 211816, China
e-mail: xn0anz@163.com

ABSTRACT Text is an essential means for humans to acquire information and engage in social communication. Accurate text extraction from images is crucial for various tasks in real-life scenarios and scene understanding. However, text detection and recognition in natural scenes are challenged by noise in the images, irregular distribution of text fonts, and degradation of image quality under complex acquisition conditions. These factors severely impact the accuracy of text recognition. Issues such as poor image quality, diverse text formats, and complex image backgrounds significantly affect the accuracy of the recognition, and these challenges remain urgent to be addressed in the field. To address these challenges, this paper proposes a transformer-based scene image text detection and recognition algorithm within a multi-scale end-to-end framework. Firstly, by integrating detection and recognition stages into an end-to-end framework, the process is simplified, reducing computation and errors. Subsequently, multi-scale characteristics are incorporated to effectively capture text information at various scales, enhancing recognition accuracy and robustness through feature fusion and anti-interference capability. Lastly, leveraging the transformer framework, the algorithm efficiently handles text information of different scales and positions, improving generalization ability. The self-attention mechanism, multi-layer stacking structure, and positional encoding in the transformer framework contribute to its effectiveness in processing diverse text information. Through validation, the proposed method demonstrates improved efficiency in scene text detection and recognition.

INDEX TERMS Text detection, text recognition, transformer, end-to-end, multi-scale.

I. INTRODUCTION

Text, as the embodiment of human wisdom, plays an indispensable role in cultural inheritance. Its emergence breaks the temporal and spatial limitations of spoken language, providing a more powerful carrier for the dissemination of human civilization. With the rapid development of information technology, the presentation of text has extended beyond the confines of traditional paper documents. A vast amount of text is now stored in the forms of documents, images, or video data. Therefore, the utilization of computer technology for scene text image detection [1] and end-to-end recognition has become particularly crucial. The detection and end-to-end recognition of text in natural scenes have extensive applications. On one hand, it can enhance the efficiency of various application scenarios, such as license plate recognition and localization, text-based captcha recognition, or handwritten text recognition [2]. On the other hand, it provides additional information in practical computer vision applications like intelligent transportation systems, image and video retrieval, guidance for visually impaired individuals, and portable visual systems. As information technology advances, the modes of text representation continue to evolve, and the significance of scene text detection and end-to-end recognition in facilitating communication and access to information is ever more pronounced.

The associate editor coordinating the review of this manuscript and approving it for publication was Qiang Li.

Due to the rich semantic information [3] embedded in scene text, it plays a crucial role in comprehending real-world scenes. With the continuous advancement of scene text detection techniques, achieving accurate text recognition in precise text localization is considered a highly challenging research problem. Most current efforts focus on the more effective extraction of visual features [4]. For instance, constructing more robust visual feature extraction [5] backbones and introducing text image correction mechanisms have achieved breakthrough progress in regular, clear-cut text cropped images. However, when dealing with irregular, fuzzy, or other complex text images, the effective extraction of visual features is insufficient to meet the accuracy demands in practical applications. This paper references a multiscale texture image segmentation method based on adaptive window fixation and propagation. By incorporating multiscale characteristics, text features of images are extracted at different scales. In scene text images, text scales, fonts, colors, orientations, and more vary due to diverse environments. Hence, extracting features solely at a single scale might not adequately handle these variations. Different scales in images allow for the use of varying receptive fields and convolutional kernels to extract features. Typically, this can be achieved by utilizing sliding windows of different sizes or employing convolutional layers of varying depths for multiscale feature extraction.

To construct a more accurate and efficient recognition network, this study draws inspiration from human understanding patterns, and introduces a novel hierarchical self-attention encoder for scene text recognition tasks. This encoder combines effective sequence semantic information with visual perception information to infer complete textual content. By employing depth-wise separable convolutions in conjunction with deep self-attention mechanisms, the study enhances the capture of correlations between visual perception and textual sequences, resulting in more robust recognition outcomes. The research integrates a convolutional neural network [6] with multiscale feature [7] fusion and the Transformer architecture into an end-to-end structure, thereby enhancing the model's generalization capability.

The contribution points of this article are as follows:

(1) Unlike traditional recurrent neural networks, the Transformer model excels in handling long text sequences through its effective self-attention mechanism. This feature not only improves the model's recognition of lengthy text but also enhances interpretability. The Transformer model's outstanding performance on benchmark datasets further underscores its advantages.

(2) By integrating multi-scale features, our model extracts text information across different scales in scene images, accommodating variations in scale, font, color, and text orientation in diverse environments. This approach, achieved through distinct-sized sliding windows or convolutional layers, enhances accuracy and robustness in scene text recognition. It effectively captures shape and structural details at different scales, proving particularly adept at handling diverse text forms. Additionally, the inclusion of multi-scale features helps address challenges like text overlap, misalignment, and skew in scene contexts.

(3) The end-to-end framework combines the detection and recognition stages, streamlining the process and reducing computation and errors. By training the detection and recognition tasks as a unified entity, the semantic information of text within images is better preserved, leading to improved training outcomes and accuracy. This approach enhances robustness, enabling better handling of text variations and noise in the scene.

The logical structure of this article is as follows:

In the second section, we present the related work of this paper and analyze various aspects of deep learning end-to-end recognition, transformer-based scene recognition, and other hierarchical attention mechanisms in text recognition. In the third section, we introduce the algorithms employed in this study and provide an overall algorithmic flowchart. The fourth section describes the experimental process, including comparative experiments and visual displays. In the fifth section, we engage in a discussion about the paper, exploring both the strengths and weaknesses of the proposed model, while also highlighting the limitations of our approach. Finally, in the sixth section, we summarize the entire research and provide prospects for future work.

II. RELATED WORK

Scene text detection is the process of locating and localizing text appearing in natural images. With the advancement of technology [39], its techniques have made significant strides, accompanied by a continuous influx of academic research. Some of these papers represent the latest achievements in scene text detection research. The development of deep learning [28], [38] and the progress in hardware for handling vast amounts of image data have also influenced the study of scene text detection. In recent years, in deep learning-based methods for natural scene text detection [32], [36], the main approaches revolve around detecting scene text from the perspectives of regressing region proposals and classifying image pixels. In literature, the classification of natural scene text detection methods mainly falls into two categories: regression-based methods and segmentation-based methods. A more detailed and rational classification of natural scene text detection methods is presented in [8], providing a better analysis and summary of existing techniques. Consequently, this paper classifies the approaches for natural scene text detection into region proposal-based methods and semantic segmentation-based methods. Reference [9] introduced the idea of deep learning to address text detection problems and achieved promising results. It employed a sliding window strategy to aid detection and used CNN to capture text features while analyzing text saliency. Reference [10] used MSER to determine candidate objects of characters. CNN was used as a classifier to aid detection, thereby filtering out the required characters. The paper also introduced sliding window techniques for auxiliary detection.

Reference [11] proposed DeepText, a fundamental architecture based on Faster-RCNN. Reference [12] introduced the CTPN algorithm, which remains a commonly used network for text detection in OCR systems, significantly influencing the direction of subsequent text detection algorithms. CTPN introduced a novel idea, similar in concept to differentiation before integration. Reference [13] proposed the TextBoxes algorithm, which is an improvement over SSD. TextBoxes innovatively considers offset information from different feature layers and incorporates it into candidate box prediction. The post-processing method employs non-maximum suppression to remove redundant detection boxes. Offers a novel perspective in scene text recognition by utilizing a single visual model with component-level mixing, merging, and combining to effectively handle text within an image tokenization framework, achieving competitive accuracy in English and significant advancements in Chinese text recognition, while maintaining faster inference speeds compared to existing methods. Proposes a novel text detection framework utilizing discrete cosine transform (DCT) for encoding text masks into compact vectors, enhancing efficiency and accuracy, with competitive performance on challenging datasets like CTW1500 and Total-Text, potentially enhancing accuracy and robustness in detecting non-standard shaped text. Introduces the Multi-Domain Character Distance Perception (MDCDP) module, leveraging position embedding and cross-attention mechanism to fuse visual and semantic features, enabling precise alignment between features and characters in scene text recognition, thereby enhancing accuracy and adaptability across diverse text domains.

Moreover, the end-to-end framework is also a crucial component, capable of integrating the detection and recognition stages, thereby streamlining the process. Reference [14] proposed an end-to-end trainable system for irregular text detection and recognition called TextNet. This approach introduces a scale-aware attention mechanism in the backbone network to extract multi-scale image features, utilized in subsequent detection and recognition tasks. In the detection branch, TextNet directly generates quadrilateral text candidate boxes to cover text regions with various orientations and deformations. Furthermore, the authors introduced a novel perspective RoI transform layer to align quadrilateral features for subsequent text recognition, allowing for more accurate recognition of irregularly shaped text. Lastly, the aligned features are encoded by an RNN into text information, and after incorporating a spatial attention mechanism, the model outputs predicted text sequences.

The task of scene text recognition involves recognizing text images cropped from scene pictures into computer-readable character sequences. Currently, mainstream frameworks for scene text recognition consist of four main stages: preprocessing, feature extraction, sequence modeling, and prediction. In the feature extraction stage, commonly used convolutional neural networks include ResNet, VGGnet, etc., while the sequence modeling stage typically employs recurrent neural networks. The prediction stage often adopts methods based on connectionist temporal classification or attention-based approaches. In contrast to traditional recurrent neural networks, this study employs the Transformer model with self-attention mechanism for scene text recognition. The Transformer model excels in handling long text sequences' dependencies, making it better suited for recognizing lengthy text sequences. With the widespread adoption of Transformers, scholars are increasingly inclined to utilize attention mechanisms to extract rich semantic information from images. For instance, [15] proposed a bidirectional decoding Transformer decoder, and [16] combined natural language processing and computer vision models based on the Transformer framework, although this method significantly increased training costs. Existing methods like RNN or LSTM often focus on sequence-based approaches or use semantic information to supervise text recognizer training, putting excessive emphasis on visual information and being susceptible to contextual semantic influences. The self-attention framework of the Transformer avoids attention drift. Currently, incorporating additional language models has also become a research hotspot. Given that extracting semantic information from pure text is much easier than from images, this paper introduces the Multi-Feature (MF) module to fuse multi-scale features, establishing multi-feature extractors capable of extracting spatial and sequence information from initial images.

End-to-end algorithms for scene text detection and recognition mainly fall into two categories: those based on pixel-level segmentation prediction and those based on RNN sequence generation. Algorithms based on pixel-level segmentation prediction achieve text detection and recognition by predicting the foreground classification of scene text. For instance, the Text Perceptron proposed in [17] employs a segmentation-based approach for text detection, transforms irregular text into regular text using a predefined shape transformation model, and finally performs recognition through a recognition network. Compared to algorithms based on pixel-level segmentation prediction, algorithms based on RNN can better encode the relationships between text characters, thus enhancing text recognition accuracy. For example, the model proposed in [18] was among the first to apply CNN and RNN to end-to-end scene text detection and recognition tasks. In the aforementioned end-to-end algorithms for scene text detection and recognition, algorithms based on pixel-level segmentation prediction can achieve precise detection down to the pixel level. However, due to the lack of exploration of textual semantic information, their recognition accuracy is somewhat limited. On the other hand, sequence generation algorithms based on RNN decoding decode the sequence within loops, resulting in a heavy computational workload and long processing times. To enhance recognition efficiency while maintaining

recognition performance, efforts are made to strike a balance between these aspects.

III. METHODS

The overall algorithm of this study can be divided into the following three parts: (1) End-to-End Recognition, (2) Multi-Scale Fusion, and (3) Transformer Framework. In addressing scene text detection and recognition in scene images, this paper achieves the following steps: Firstly, by employing an end-to-end framework, the detection and recognition stages are merged together. Following this, the multi-scale fusion approach is applied to effectively capture text information at various scales. This facilitates enhanced recognition accuracy and robustness through feature fusion and anti-interference capabilities. Finally, leveraging the transformer framework, the model utilizes its inherent self-attention mechanism, multi-layer stacking structure, and position encoding characteristics to effectively process text information at different scales and positions, thereby enhancing generalization capability. The overall algorithm flowchart of this article is shown in Figure 1.

A. END-TO-END FRAMEWORK

This paper proposes a fusion of the adaptive Bézier curve network that can be end-to-end trained for arbitrary-shaped scene text detection and recognition tasks. The architecture is shown in Figure 2. Compared to standard rectangular bounding box-based text detection methods, ABCNet [19] achieves arbitrary-shaped scene text detection through simple yet effective adaptive Bézier curve bounding boxes with almost no additional computational overhead, greatly simplifying the complexity of the recognition branch. In contrast to existing end-to-end scene text detection and recognition methods, ABCNet employs parameterized Bézier curves to describe arbitrary-shaped text, significantly enhancing the overall framework's operational efficiency while ensuring algorithmic detection and recognition performance. Its computational formula is as follows:

ABCNet uses Bernstein polynomials as the basis for the Bessel parameter curve $c(t)$:

$$c(t) = \sum_{i=0}^{n} b_i B_{i,n}(t), 0 \le t \le 1 \tag{1}$$

Among them, n represents the angle, b_i represents the i-th control point, and $B_{i,n}(t)$ represents the Bernstein polynomial.

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, i = 0, \ldots, n \tag{2}$$

Among them, $\binom{n}{i}$ is the binomial coefficient. By observing existing datasets and curve texts in practical applications, ABCNet adopts a cubic Bessel curve, where n is 3.

For each text instance, ABCNet uses the relative distance as the regression target as follows:

$$\Delta_x = b_{ix} - x_{\min}, \Delta_y = b_{iy} - y_{\min} \tag{3}$$

where x_{min} and y_{min} represent the minimum x and y values of the 4 vertices, respectively.

To train the model for the task of predicting control point coordinates, it is necessary to generate Bézier curve labels and then follow a method similar to DMPNet to regress the target coordinates. The advantage of using predicted relative distances as regression targets is that accurate predictions can still be made when Bézier control points extend beyond the image boundaries. Within the detection branch, only a convolutional layer with 16 output channels is required to learn the prediction of both Δx and Δy. This enables the detection branch to accurately output prediction results with almost no additional computational overhead.

To generate Bezier curve labels using the original polygon annotation, let $(p_i)_{i=1}^{n}$ be a set of polygon boundary annotation points, where p_i represents the i-th annotation point. Simply apply the standard least squares method to obtain the optimal parameters of $c(t)$ under the cubic Bezier curve in the formula to generate the Bezier curve boundary box:

$$\begin{bmatrix} B_{0,3}(t_0) & \cdots & B_{3,3}(t_0) \\ \vdots & \ddots & \vdots \\ B_{0,3}(t_m) & \cdots & B_{3,3}(t_m) \end{bmatrix} \begin{bmatrix} b_{x0} & b_{y0} \\ b_{x1} & b_{y1} \\ b_{x2} & b_{y2} \\ b_{x3} & b_{y3} \end{bmatrix} = \begin{bmatrix} p_{x0} & p_{y0} \\ p_{x1} & p_{y1} \\ \vdots & \vdots \\ p_{xm} & p_{ym} \end{bmatrix} \tag{4}$$

Among them, m represents the number of points labeled by the curved boundary. For the two commonly used curve scene text detection and recognition datasets Total Text and CTW1500, the values of m are 5 and 7, respectively.

By using the ratio of accumulated length to polyline perimeter, the parameter "t" can be calculated. According to the formula mentioned above, the original polyline annotations can be transformed into parameterized Bézier curves. It's important to note that the first and last annotated points will be used as the first "(b_0)" and last "(b_4)" Bézier curve control points, respectively. Bounding boxes generated through the above method of Bézier curves often exhibit better visual results compared to the original polygon annotations. Additionally, by utilizing the proposed feature alignment method and the generated Bézier curve bounding boxes, it becomes easy to transform curved scene text into horizontal text without significant distortion. The simplicity of this method allows for its application to different types of text in practical scenarios.

B. MULTISCALE FUSION

Multi-scale feature fusion [20] can enhance the classification performance of a network by fusing feature maps from different scales. Common multi-scale feature fusion networks are mainly divided into parallel multi-branch networks and sequential skip connection structures. In this study, as the

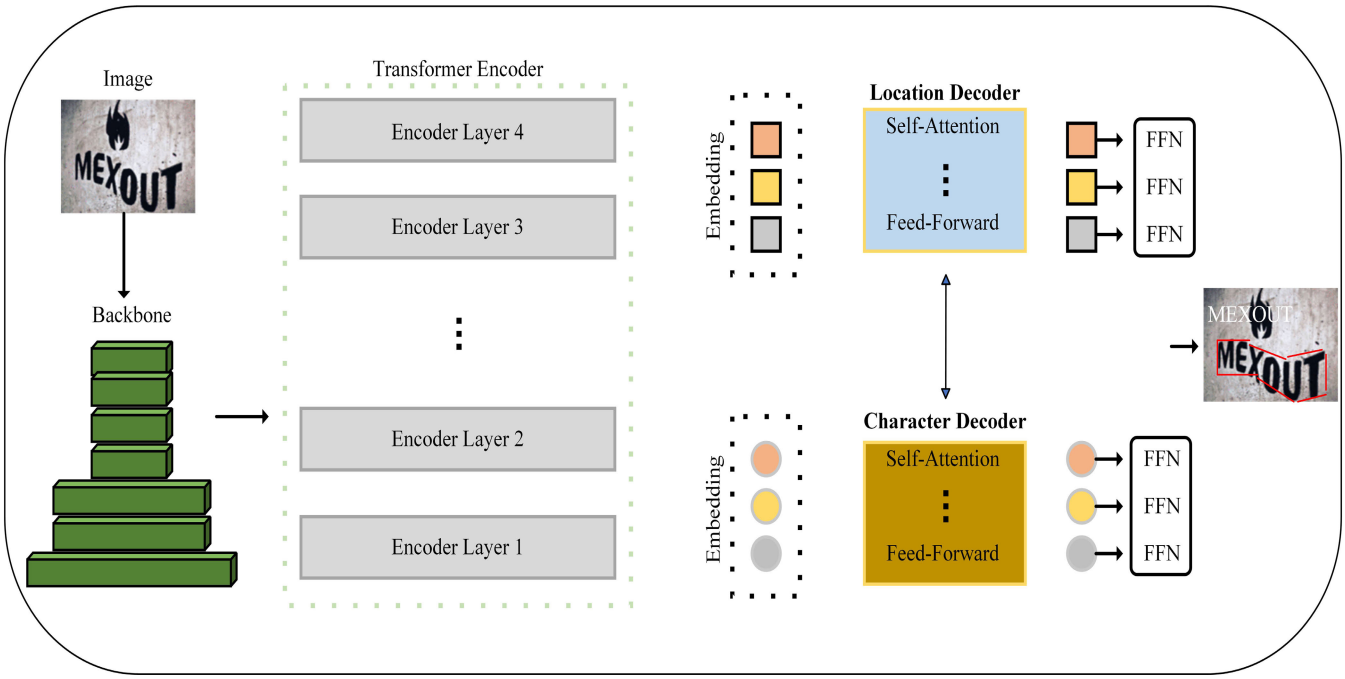

FIGURE 1. Algorithm flowchart.

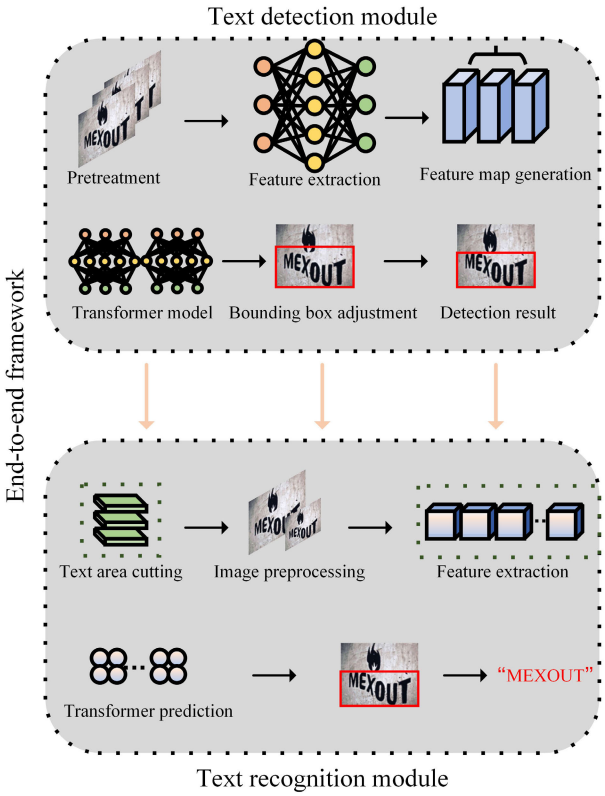

FIGURE 2. The architecture.

multi-scale fusion dataset used is processed end-to-end and the annotated samples are divided into two classes, abnormal regions and normal regions, employing networks that are too deep or overly complex can lead to overfitting issues. Therefore, based on a streamlined approach, this research proposes a scene text recognition method using a multi-scale fusion feature network, as depicted in Figure 3. Initially, text images are preprocessed and resized to 48 × 48 grayscale images. Then, these preprocessed images are fed into the multi-scale fusion feature network for feature extraction. Finally, Softmax classification is performed. An Inception structure containing convolutional layers with different scales is utilized to obtain local detailed features from varying receptive fields. The outputs of block2, block3, and block5 are employed as the final fused text image feature maps.

The process of complex scene text recognition is intricate and often requires extensive labeled data, leading to lengthy training times. Thus, the utilization of regularization methods becomes crucial to reduce network complexity, prevent overfitting, and enhance model generalization. In this study, three regularization methods - batch normalization, L2 regularization, and dropout - were employed within the multi-scale fusion network to improve model generalization. These methods collectively serve to enhance the ability of the model to generalize beyond the training data and mitigate overfitting issues.

Utilizing L2 regularization involves adding a regularization term to the loss function. This term penalizes larger weight values, causing all weights to converge towards smaller absolute values. This serves the purpose of guiding and influencing network training through the regularization term. The formula for L2 regularization is derived as follows:

$$J = J_0 + \frac{\lambda}{2m} \sum_{i=1}^{n} w_i^2 \tag{5}$$

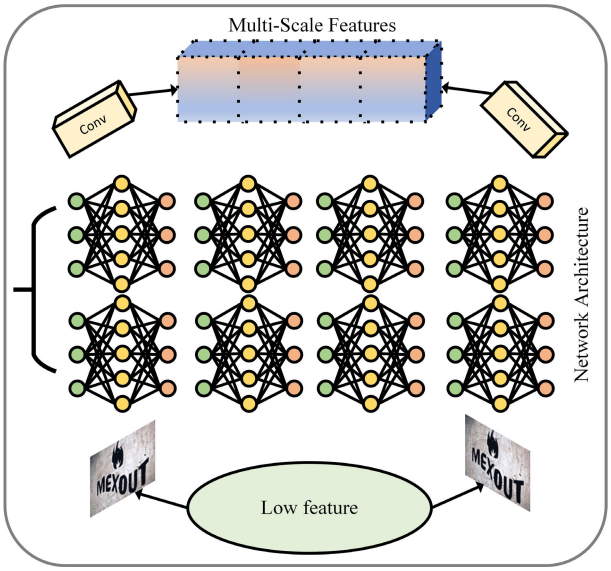

FIGURE 3. A scene text recognition method using a multi-scale fusion feature network.

Among them, J_0 represents the original loss function, $\sum_{i=1}^{n} w_i^2$ is the L2 regularization term, λ is the regularization coefficient, m is the number of training samples for a single batch, and the function of coefficient 1/2 is to simplify the differentiation process. Adding a dropout layer after the fully connected layer causes the activation value of a neuron to stop working with a certain probability in forward propagation, reducing the interaction between hidden nodes. Based on experience, θ is set to 0.3.

Constructing the spatial and channel attention modules involves deriving relationships between specific information across pixels in terms of spatial relations and interdependencies among feature channels. This process begins by reshaping the input features and calculating a spatial similarity matrix. Then, a channel similarity matrix is computed using matrix multiplication. The Softmax function is utilized to generate the spatial attention weights. The specific formula is as follows:

$$p_{ij} = \frac{\exp\left(A_i^T \times A_j\right)}{\sum_{i=1}^{N} \exp\left(A_i^T \times A_j\right)} \tag{6}$$

$$q_{ij} = \frac{\exp\left(A_i \times A_j^T\right)}{\sum_{i=1}^{C} \exp\left(A_i \times A_j^T\right)} \tag{7}$$

where p_{ij} represents the impact of the i-th position on the jth position, q_{ij} represents the impact of the i-th channel on the jth channel, A_i^T represents the i-th row of A^T, and A_j represents the jth column of A. Fusion channel and spatial attention mechanism, calculate the feature information of fused spatial attention through matrix multiplication between A and P, and obtain the feature information using pixel addition method:

$$F_{\text{refine}} = \text{reshape}\ (A \times P_{\text{att}}) \oplus \text{reshape}\ (Q_{\text{att}} \times A) \tag{8}$$

Among them, F_{refine} represents the feature information of the attention mechanism module, reshape the feature information with dimension size $C \times N$ to $C \times H \times W$, P_{att} is the spatial attention weight generated by the Softmax function, and Q_{att} is the channel attention weight generated by the Softmax function.

Furthermore, in the process of multi-scale fusion, to expand the receptive field while keeping the relative spatial positions of pixels unchanged and gaining more contextual information for the segmentation task, this paper incorporates dilated convolutions. Unlike standard convolutions, dilated convolutions introduce an additional parameter known as the dilation rate. The dilation rate controls the spacing between adjacent elements in the convolution kernel. As the dilation rate changes, the size of the convolution kernel's receptive field also changes. An example of this is a convolution with a size of 3×3, a dilation rate of 2, and a stride of 1, referred to as dilated convolution. In this study, the convolutional layers of VGG16 are replaced with dilated convolutional layers with a dilation rate of 2. This modification enhances the receptive field of the main feature extraction network without increasing computational complexity.

C. TRANSFORMER MODEL

In this paper, by utilizing the end-to-end framework of ABCNet, the merging of detection and recognition [21], [22] yields promising results. Subsequently, by integrating multi-scale features, further improvements are achieved in terms of recognition accuracy and robustness. The final proposed scene text recognition model, based on CNN and Transformer, is illustrated in Figure 4. This model comprises three main components: the CNN feature extraction layer, the Transformer encoder, and the CTC decoder.

We have constructed a representation method suitable for text image information feature extraction, using a combination of word2vec and CNN to extract semantic features of scene images, ultimately forming a feature representation of text. It includes three layers, namely input layer, projection layer, and output layer. Among them, the input layer inputs the current feature text, and the word vector $W_t \in R^m$ of the text; The output is the probability of words appearing in the context of the feature word; The purpose of the projection layer is to maximize the L-value of the objective function.

$$L = \frac{1}{N} \sum_{j=1}^{N} \sum_{-c \le i \le c} logp\left(\omega_{j+1} | \omega_j\right) \tag{9}$$

Among them, N is the length of the word sequence, c is the contextual length of the current feature word, and $p\left(\omega_{j+1} | \omega_j\right)$ is the probability of the contextual feature word ω_{j+1} appearing when the current word ω_j is known. All word vectors obtained through model training form a word vector matrix $X \in R^{mn}$, where $x_i \in R^m$ represents the word vector of the feature word i in the m-dimensional space.

Convolutional layer is the most important layer in CNN, with two key operations: local correlation and window

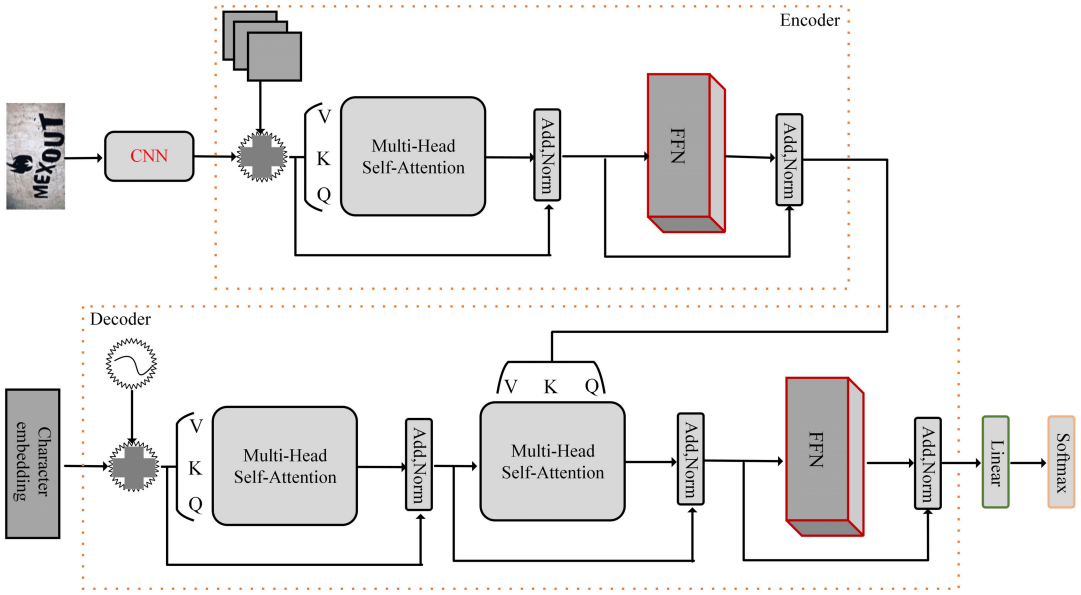

FIGURE 4. Scene text recognition mode.

sliding. The input in the convolutional layer is a matrix with dimension m times n, which is used to represent the sentence matrix of compilation error information. When performing convolution operations in the convolutional layer, the selection of the width of the convolutional kernel is consistent with the dimension of the word vector, which ensures that the convolutional kernel is a complete word vector at each sliding position. The convolution calculation formula is as follows:

$$y_i = f\left(\sum w_i \times x_{i:i+h-1} + b\right) \tag{10}$$

where, w_i represents the weight matrix of the convolutional kernel, $x_{i:i+h-1}$ denotes the matrix of word vectors from the i-th row to $i+h-1$ rows, b stands for bias, and the function f signifies the activation function.

The Transformer consists of two sub-layers: a Multi-Head Attention (MHA) layer and a Multi-Layer Perceptron (MLP) layer. Before each sub-layer, Layer Normalization (LN) is applied, and residual connections are used after each sub-layer. The calculations are as follows:

$$head_j = Attention(QW_j^Q, Avg\ (K)\, W_j^K, Avg\ (V)W_j^V) \tag{11}$$

Among them, W_j^Q, W_j^K, and W_j^V represent the weight matrix corresponding to the jth header and input, W^o represents the weight matrix of the linear layer, and $Avg\ (\cdot)$ represents the average pooling, with the aim of reducing parameter computation. $Attention(\cdot)$ is calculated as follows:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_{head}}}\right) V \tag{12}$$

Perform a dot product between the query matrix Q and the key matrix K, then divide the result by $\sqrt{d_{head}}$, where d_{head} represents the dimension of each head (used to balance gradient changes). Normalize the dot product result using the Softmax function to obtain attention weights, indicating the level of association between queries and keys. Proceed to compute the dot product between the normalized attention weights and the value matrix V, resulting in the final output. The algorithm pseudocode in this article is shown in Algorithm 1.

IV. EXPERIMENT

A. EXPERIMENTAL ENVIRONMENT

This paper presents a Transformer-based approach for scene text detection and recognition. The experimental setup is as follows: The model is built using the PyTorch framework and consists of four main parts, including multi-scale feature extraction, feature fusion, text detection, and text recognition. Text detection employs an attention-based end-to-end text detection model, while text recognition utilizes a Transformer-based recognition model. Finally, evaluation metrics include the F1 score for text detection and accuracy for text recognition. Experimental results demonstrate that the proposed method exhibits excellent performance in scene text detection and recognition tasks. The experimental flowchart of this article is shown in Figure 5.

B. EXPERIMENTAL DATASET

In experiments, training sets are usually used for model training, and test sets are used for model evaluation. The evaluation indicators include: accuracy of text detection, recall rate, F1 value, and accuracy of text recognition. In order to improve the robustness of the model, cross validation technology is used to evaluate the performance of the model. Based on the description of the dataset, in order to train and evaluate the performance of scene text detection and

Algorithm 1 ETE-MSFT Network Training

Inputs: Training data from COCO-Text, SynthText, ICDAR 2017, and Street View Text dataset

Outputs: Trained ETE-MSFT network parameters

Initialize ETE-MSFT network with Transformer architecture Initialize attention mechanism parameters Initialize multi-scale features extractor parameters Initialize learning rate, batch size, and other hyperparameters

while *not converged* **do**

  for *each batch of training samples* **do**

    Calculate multi-scale features for the batch using the feature extractor Calculate attention weights for each feature map using the attention mechanism Apply the attention weights to the features to obtain attended features Apply the Transformer encoder on the attended features to learn contextual representations Predict the text localization using a regression head Compute the text recognition loss using CTC loss for text transcription Compute the localization loss using bounding box regression Calculate the total loss as a combination of recognition and localization losses Update the network parameters using backpropagation and optimizer

  Calculate Precision(%), Recall(%), and F1(%) using validation dataset Calculate the current FPS and update the best FPS achieved so far **if** *current F1(%) is higher than the best F1(%)* **then**

    Save the current ETE-MSFT model as the best model

  if *no significant improvement in validation F1(%) for certain epochs* **then**

    Reduce learning rate with a certain decay factor

Return: Trained ETE-MSFT network with the best model parameters

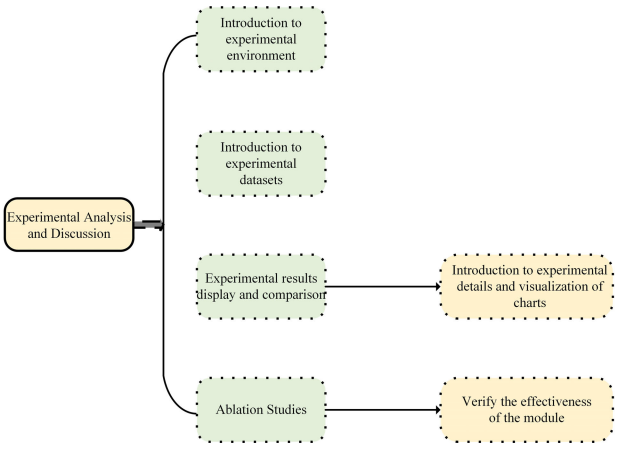

FIGURE 5. Experimental flow chart.

recognition models, the following four sets of datasets were selected:

(1) SCUT CTW1500: The dataset includes 1000 images for training and 500 images for testing. The text lines in each image are described using a polygon composed of 14 points. This dataset is mainly aimed at curve shaped text and is one of the few datasets in the field of text detection in arbitrary shaped scenes.

(2) ICDAR2017: This dataset contains text of various shapes and directions, including horizontal text, vertical text, curved text, rotated text, arbitrarily shaped text, etc. In addition, the dataset also provides text data in multiple languages, including English, Chinese, Arabic, etc. Therefore, this dataset is more comprehensive and challenging, and has more reference value for evaluating and comparing the performance of various character and text recognition algorithms.

(3) SVT: This dataset is collected in street scenes, and there are widespread cases of blurring, noise, and low resolution in the images. This dataset is of great significance for testing the robustness and stability of algorithms in real-world scenarios, therefore it is also a very useful dataset.

(4) Synth Text: This dataset includes 800000 synthesized images, mainly used for training in scene text recognition. Synthetic datasets can provide a large amount of training data, which can improve the generalization ability and robustness of the model. Therefore, Synth Text is a very useful dataset, especially in the research of scene text recognition.

C. EXPERIMENTAL EVALUATION INDICATORS

The evaluation indicators for text detection include precision, recall, and F1 value. Accuracy represents the proportion of detected text regions that are truly text, recall represents the proportion of detected text regions, and F1 value is the harmonic average of accuracy and recall, which is an important indicator for evaluating the performance of detection models. Model computational complexity: The computational complexity of a model is an important indicator for evaluating model performance, including the number of parameters, computational complexity, and inference time of the model. Robustness: The robustness of a model is one of the important indicators for evaluating its performance, including its stability and transferability in different datasets and environments.

The formula is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

Among them, TP stands for True Positive, that is, the number of correctly identified entities; FP stands for False Positive, that is, the number of non-entities incorrectly recognized.

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

Among them, FN represents False Negative, that is, the number of real entities that are missed. In addition, there is the calculation of F1 value:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{15}$$

These indicators comprehensively consider the detection and recognition performance, computational efficiency, and robustness of the model, and can comprehensively evaluate the performance and practicality of Transformer based scene image text detection and recognition methods. In experiments, techniques such as cross validation are usually used to evaluate the performance of the model to ensure the reliability and stability of the experimental results.

D. ANALYSIS OF EXPERIMENTAL DETAILS

The Transformer-based scene text detection and recognition method has been studied and applied for recognizing scene text in different orientations, including rotated text. Below, we will provide a detailed overview of comparative experiments of the Transformer-based scene text detection and recognition method on scene text in various orientations, including rotated text.

In Table 1, we conducted a comparative analysis of the performance of different text detection and recognition methods on the COCO Text dataset and the SynthText dataset. The following provides a more detailed explanation of these data to highlight the superiority of our method. On the COCO-Text dataset, our method showed significant performance improvement compared to other methods. In terms of accuracy, our method achieved 94.28%, which is higher than other methods, with a maximum improvement of about 12 percentage points. In terms of recall rate, our method reached 93.84%, which is also higher than other methods, increasing by about 13 percentage points. In terms of F1 value, our method reached 94.05%, which is also significantly higher than other methods, increasing by about 13 percentage points. This demonstrates the significant superiority of our method on the COCO Text dataset; On the SynthText dataset, our method also demonstrated excellent performance compared to other methods. In terms of accuracy, our method achieved 93.37%, which is higher than other methods, with a maximum improvement of about 11 percentage points. In terms of recall rate, our method reached 91.46%, which is also higher than other methods, increasing by about 15 percentage points. In terms of F1 value, our method reached 92.40%, which is also significantly higher than other methods, increasing by about 14 percentage points. This further highlights the excellent performance of our method on the SynthText dataset. In summary, our method achieved significant performance improvements on two different datasets, with an improvement of multiple percentage points compared to other methods, demonstrating the significant superiority of our method in scene text detection and recognition tasks. We compared and visualized the results in Table 1, as shown in Figure 6.

According to Table 2, we analyzed the performance of different text detection and recognition methods on the ICDAR 2017 dataset and the Street View Text dataset. Our method (''Ours'') demonstrated excellent performance on the ICDAR 2017 dataset. In terms of accuracy, our method achieved 92.91%, which is higher than other methods, with a maximum improvement of about 9 percentage points. In terms of recall rate, our method reached 89.77%, which is also higher than other methods, increasing by about 6 percentage points. In terms of F1 value, our method reached 91.31%, which is significantly higher than other methods and has increased by about 7 percentage points. This indicates that our method exhibits significant superiority on the ICDAR 2017 dataset; On the Street View Text dataset, our method also demonstrated excellent performance compared to other methods. In terms of accuracy, our method achieved 94.66%, which is higher than other methods, with a maximum improvement of about 11 percentage points. In terms of recall rate, our method reached 93.49%, which is also higher than other methods, increasing by about 12 percentage points. In terms of F1 value, our method achieved 94.07%, which is significantly higher than other methods and has increased by about 10 percentage points. This once again highlights the excellent performance of our method on the Street View Text dataset. Overall, our method achieved significant performance improvements on both the ICDAR 2017 dataset and the Street View Text dataset, with an improvement of multiple percentage points compared to other methods, demonstrating the significant superiority of our method in different datasets and scenarios. We compared and visualized the results in Table 2, as shown in Figure 7.

In Table 3, we conducted a detailed analysis of the parameter size and frame rate (FPS) of different text detection and recognition methods on the COCO Text dataset and SynthText dataset, and compared them with other methods. Compared to other methods, our method exhibits significant advantages on the COCO Text dataset. Specifically, our method has a parameter count of only 41.92M, which is significantly lower than other methods, with the highest method having a parameter count of 83.64M. This means that our method is lighter in model size, which helps reduce storage and computational costs. In addition, our method

TABLE 1. Performance comparison of text detection and recognition methods on COCO text and SynthText datasets.

Method	COCO-Text dataset [30]			SynthText dataset [31]		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Manjari et al. [23]	81.36	79.84	80.59	79.65	75.84	77.69
Prabu et al. [24]	82.57	80.65	81.59	81.59	80.38	80.98
Larbi et al. [25]	76.89	77.98	77.43	88.67	87.69	88.17
Tarride et al. [26]	75.41	76.32	75.86	77.16	74.76	75.94
Bhatt et al. [27]	89.63	87.81	88.71	88.67	85.84	87.23
Vishwakarma et al. [29]	91.37	86.29	88.75	92.61	90.58	91.58
Ours	**94.28**	**93.84**	**94.05**	**93.37**	**91.46**	**92.40**

TABLE 2. Performance analysis of text detection and recognition methods on ICDAR 2017 and street view text datasets.

Method	ICDAR 2017 dataset [33]			Street View Text dataset [35]		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Manjari et al. [23]	73.68	74.63	74.15	78.53	76.41	77.45
Prabu et al. [24]	85.37	83.47	84.40	80.28	79.89	80.08
Larbi et al. [25]	78.64	76.32	77.46	89.31	85.46	87.34
Tarride et al. [26]	90.39	84.12	87.14	85.13	81.68	83.36
Bhatt et al. [27]	83.68	80.95	82.29	83.55	83.04	83.29
Vishwakarma et al. [29]	87.34	84.38	85.83	79.37	78.09	78.72
Ours	**92.91**	**89.77**	**91.31**	**94.66**	**93.49**	**94.07**

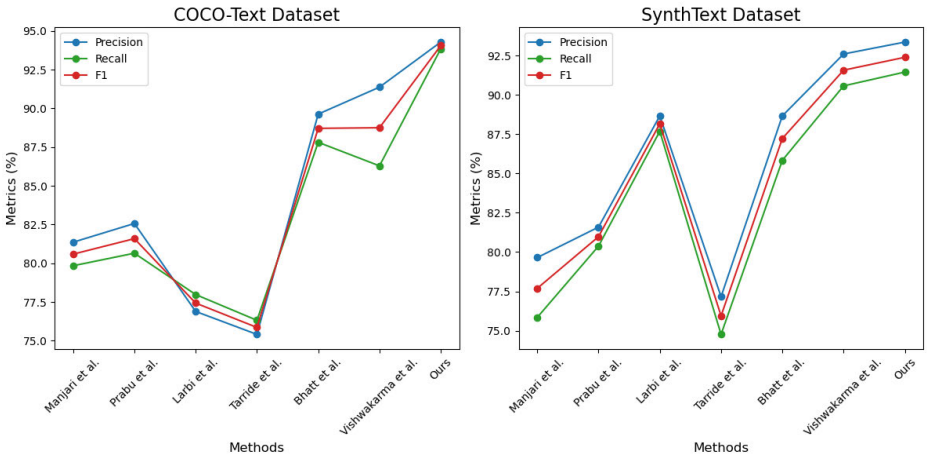

FIGURE 6. Performance comparison of text detection and recognition methods on COCO text and SynthText datasets.

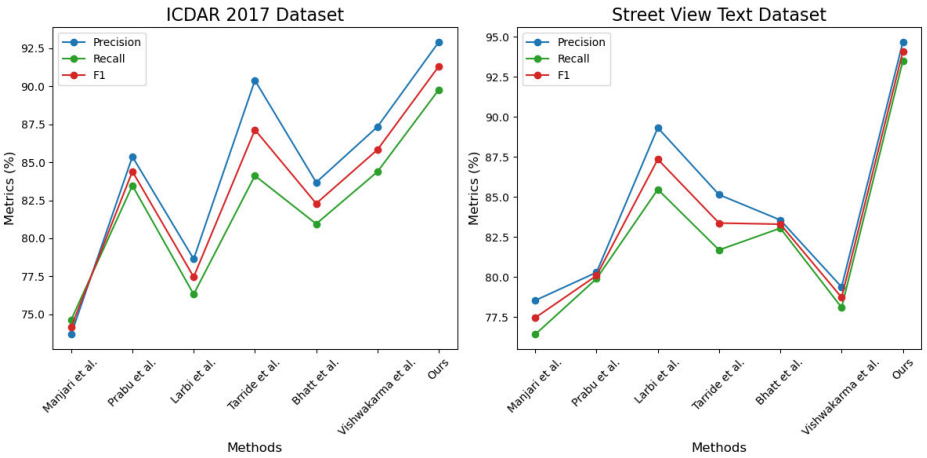

FIGURE 7. Performance analysis of text detection and recognition methods on ICDAR 2017 and street view text datasets.

achieved the highest frame rate on the COCO Text dataset, at 68.89 FPS, while other methods had frame rates ranging from 30 FPS to 52.73 FPS. This indicates that our method is not only more efficient in model size, but also able to handle text detection and recognition tasks faster; On the SynthText dataset, our method also exhibits significant advantages over other methods. Our method has a parameter size of 42.15M, which is relatively low, while the frame rate is 60.93 FPS, which is higher than other methods. Compared with other methods, our method has competitive advantages in model size and processing speed. In summary, our method has lower parameter count and higher frame rate compared to other methods on the COCO Text dataset and SynthText dataset. These results clearly demonstrate the excellent performance and efficiency of our method in large-scale text detection and recognition tasks, providing strong support for practical applications. We compared and visualized the results in Table 3, as shown in Figure 8.

In Table 4, we conducted a detailed analysis of the parameter size and frame rate (FPS) of different text detection and recognition methods on the ICDAR 2017 dataset and Street View Text dataset, and compared them with other methods. On the ICDAR 2017 dataset, our method showed significant competitive advantages in terms of parameter size and frame rate. Our method has a parameter size of 45.89M, which is relatively low, and a frame rate of 62.72 FPS, which is higher than other methods. In contrast, the parameter range of other methods is between 54.36M and 76.35M, and the frame rate range is between 30.89 FPS and 58.23 FPS. This

TABLE 3. Parameter size and FPS analysis of text detection and recognition methods on COCO text and SynthText datasets.

Method	COCO-Text dataset [30]		SynthText dataset [31]	
	Parameter(M)	FPS	Parameter(M)	FPS
Manjari et al. [23]	83.64	52.73	53.86	31.78
Prabu et al. [24]	76.35	36.88	62.04	57.15
Larbi et al. [25]	49.86	29.67	51.57	42.32
Tarride et al. [26]	51.03	43.05	67.91	38.47
Bhatt et al. [27]	46.38	38.71	48.80	50.26
Vishwakarma et al. [29]	68.07	30.56	57.42	53.74
Ours	**41.92**	**68.89**	**42.15**	**60.93**

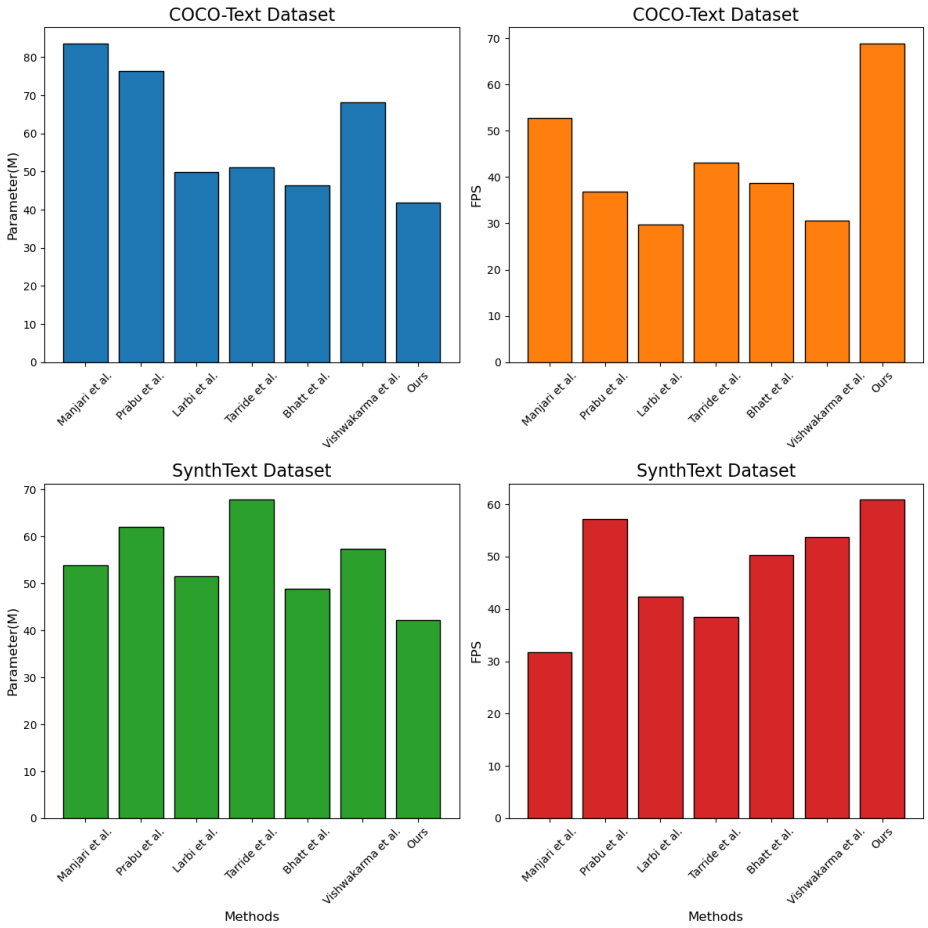

FIGURE 8. Parameter size and FPS analysis of text detection and recognition methods on COCO text and SynthText datasets.

indicates that our method is competitive in both model size and processing speed, and can more efficiently handle text detection and recognition tasks in the ICDAR 2017 dataset; Our method also demonstrated superior performance on the Street View Text dataset. Our method has a parameter size of 41.94M, which is relatively low, while the frame rate is 62.45 FPS, which is higher than other methods. The parameter range of other methods is between 45.78M and 65.12M, and the frame rate range is between 31.81 FPS and 53.09 FPS. This further highlights the excellent performance of our method in terms of model size and processing speed. In summary, our method has lower parameter count and higher frame rate compared to other methods on the ICDAR 2017 dataset and Street View Text dataset. These results further demonstrate the excellent performance and efficiency of the proposed method on different datasets, indicating its practical application potential in various text detection and recognition tasks. We compared and visualized the results in Table 4, as shown in Figure 9.

In Table 5, we conducted a series of ablation experiments by gradually adding different modules to assess their impact on model performance. These experiments were performed

TABLE 4. Parameter size and FPS analysis of text detection and recognition methods on ICDAR 2017 and street view text datasets.

Method	ICDAR 2017 [33]		Street View Text dataset [35]	
	Parameter(M)	FPS	Parameter(M)	FPS
Manjari et al. [23]	64.23	36.36	49.86	50.18
Prabu et al. [24]	76.35	48.07	65.12	35.72
Larbi et al. [25]	56.42	43.42	53.07	53.09
Tarride et al. [26]	72.07	30.89	59.51	48.96
Bhatt et al. [27]	54.36	58.23	45.78	31.81
Vishwakarma et al. [29]	66.19	53.19	61.23	44.29
Ours	**45.89**	**62.72**	**41.94**	**62.45**

using the COCO-Text dataset and the SynthText dataset, and measured precision (P), recall (R), and frames per second (FPS). First, we can observe the performance of the baseline model, which achieved precision and recall between 79.31% and 78.68% (COCO-Text dataset), as well as 78.28% and 77.57% (SynthText dataset). The FPS was 45.13 (COCO-Text dataset) and 47.92 (SynthText dataset). Next, by adding multi-scale feature fusion (+MS), the model's performance improved on both datasets. Precision and recall increased to 83.29% and 82.03% (COCO-Text dataset), as well as 85.34% and 84.63% (SynthText dataset). The FPS also slightly improved to 51.72 (COCO-Text

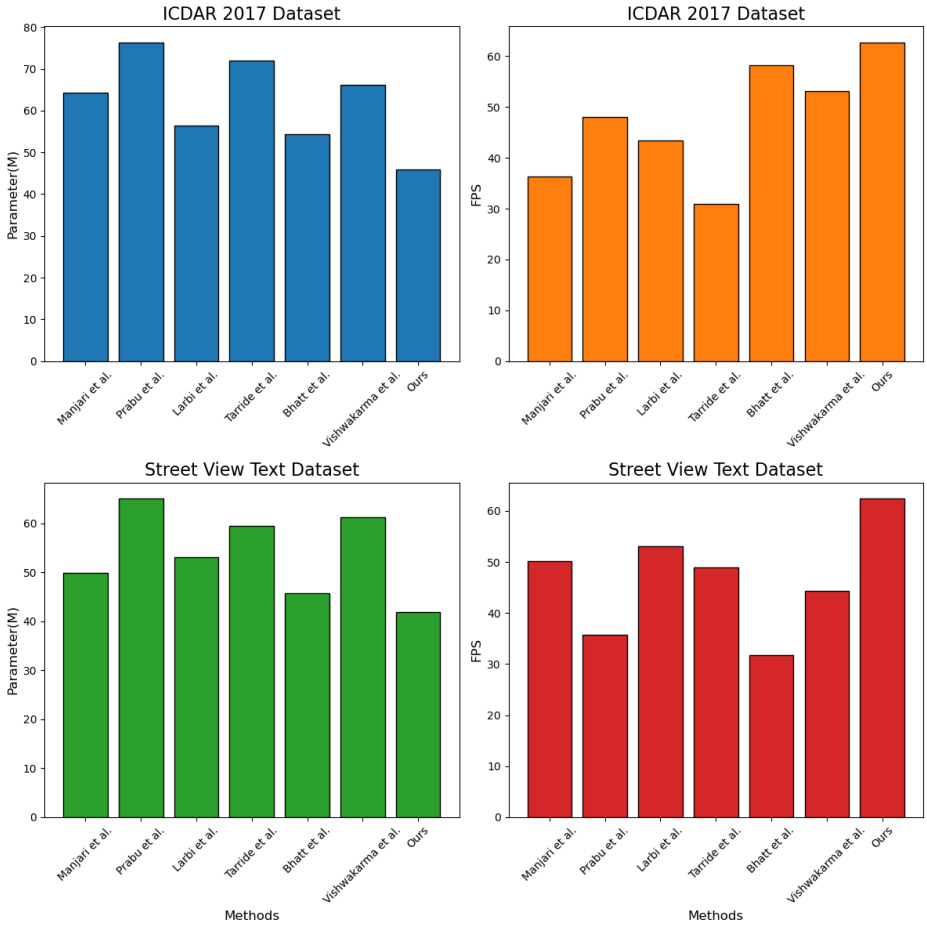

FIGURE 9. Parameter size and FPS analysis of text detection and recognition methods on ICDAR 2017 and street view text datasets.

dataset) and 51.46 (SynthText dataset). Subsequently, adding end-to-end training (+EtE) further enhanced the model's performance. On the COCO-Text dataset, precision and recall reached 87.4% and 86.94%, respectively, with an FPS of 56.89. On the SynthText dataset, precision and recall reached 90.29% and 88.74%, respectively, with an FPS of 55.82. Finally, combining multi-scale feature fusion and end-to-end training (+MS EtE), the model achieved the highest level of performance on both datasets. On the COCO-Text dataset, precision and recall reached 94.28% and 93.84%, respectively, with a significantly improved FPS of 68.89. On the SynthText dataset, precision and recall reached 93.37% and 91.46%, respectively, with an FPS of 60.93. We have compared and visualized the results of Table 5 as shown in Figure 10.

In Table 6, we present the results of a series of ablation experiments conducted on the ICDAR 2017 dataset and the Street View Text dataset. These experiments measured precision (P), recall (R), and frames per second (FPS). Firstly, we observed the performance of the baseline model. On the ICDAR 2017 dataset, the baseline model achieved precision and recall of 79.84% and 77.91%, respectively, with a frame rate of 46.15 FPS. On the Street View Text dataset, the

TABLE 5. Ablation experiment results: impact of multi-scale feature fusion and end-to-end training on COCO-text and SynthText datasets.

Module	COCO-Text dataset [30]			SynthText dataset [31]		
	P(%)	R(%)	FPS	P(%)	R(%)	FPS
baseline	79.31	78.68	45.13	78.28	77.57	47.92
+MS	83.29	82.03	51.72	85.34	84.63	51.46
+EtE	87.4	86.94	56.89	90.29	88.74	55.82
+MS EtE	94.28	93.84	68.89	93.37	91.46	60.93

baseline model achieved precision and recall of 78.01% and 75.3%, respectively, with a frame rate of 43.97 FPS. Next, by adding the multi-scale feature fusion module (+MS), the model's performance improved on both datasets. Precision and recall increased to 82.69% and 81.56% (ICDAR 2017 dataset), as well as 85.79% and 84.07% (Street View Text dataset). The frame rate also slightly improved to 50.78 FPS (ICDAR 2017 dataset) and 52.64 FPS (Street View Text dataset). Subsequently, the addition of the end-to-end training module (+EtE) further enhanced the model's performance. On the ICDAR 2017 dataset, precision and recall reached 88.26% and 85.09%, respectively, with a frame rate of 55.02 FPS. On the Street View Text dataset, precision

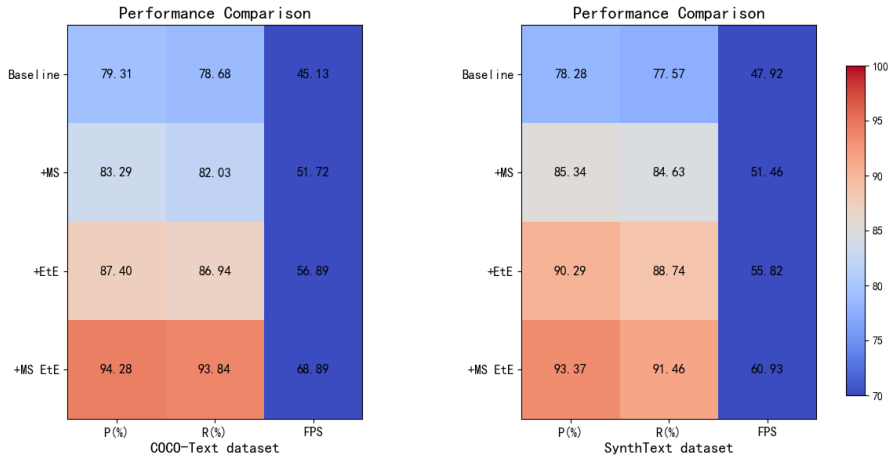

FIGURE 10. Ablation experiment results: impact of multi-scale feature fusion and end-to-end training on COCO-text and SynthText datasets.

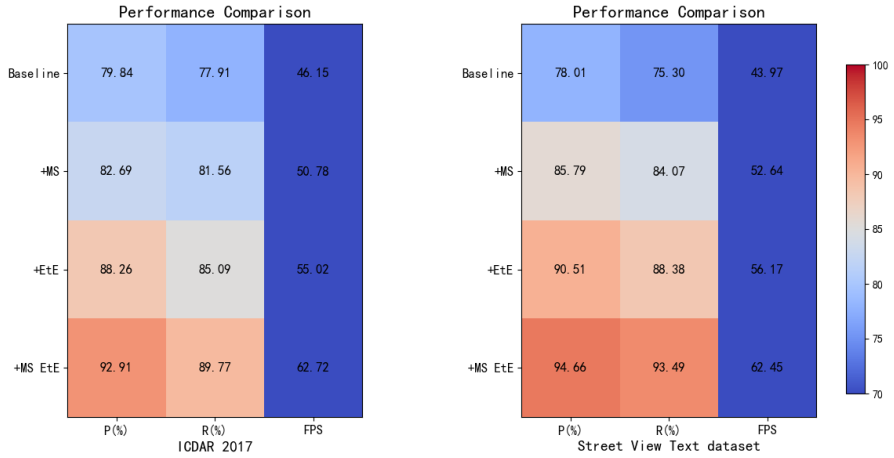

FIGURE 11. Ablation experiment results: impact of multi-scale feature fusion and end-to-end training on ICDAR 2017 and street view text datasets.

and recall reached 90.51% and 88.38%, respectively, with a frame rate of 56.17 FPS. Finally, combining the multi-scale feature fusion and end-to-end training modules (+MS EtE), the model achieved the highest level of performance on both datasets. On the ICDAR 2017 dataset, precision and recall reached 92.91% and 89.77%, respectively, with a significantly improved frame rate of 62.72 FPS. On the Street View Text dataset, precision and recall reached 94.66% and 93.49%, respectively, with a frame rate of 62.45 FPS. We have also visually compared the results of Table 6, as shown in Figure 11.

Overall, the results of the ablation experiment clearly demonstrate the significant improvement of model performance by multi-scale feature fusion and end-to-end training. Their combination has shown excellent accuracy in text detection and recognition tasks, and has also achieved significant gains in frame rate, further verifying the importance of these modules in improving model performance.

TABLE 6. Ablation experiment results: impact of multi-scale feature fusion and end-to-end training on ICDAR 2017 and street view text datasets.

Module	ICDAR 2017 [33]			Street View Text dataset [35]		
	P(%)	R(%)	FPS	P(%)	R(%)	FPS
baseline	79.84	77.91	46.15	78.01	75.3	43.97
+MS	82.69	81.56	50.78	85.79	84.07	52.64
+EtE	88.26	85.09	55.02	90.51	88.38	56.17
+MS EtE	92.91	89.77	62.72	94.66	93.49	62.45

V. DISCUSSION

In the discussion, we will focus on several key aspects. Firstly, the advantage of multi-scale feature fusion: this paper adopts multi-scale feature fusion technology, and experimental results show a significant improvement in precision and recall on multiple datasets, ultimately achieving good performance. This demonstrates that multi-scale feature fusion is an effective strategy for improving scene text detection and recognition performance. Secondly, the impact

of end-to-end training: this paper introduces an end-to-end training module, which improves model performance by simultaneously considering multiple tasks in text detection and recognition. By jointly optimizing text detection and recognition tasks, the model exhibits excellent performance on various performance metrics, further confirming the effectiveness of end-to-end training. Next, the validation of ablation experiments: in order to gain a deeper understanding of the impact of each module on performance, this paper conducts ablation experiments. The experimental results show that the gradual introduction of each module has a positive impact on performance. The combination of multi-scale feature fusion and end-to-end training modules achieves the best results on all datasets, further demonstrating their complementary roles. However, there are also limitations, such as high computational resource requirements, dataset dependencies, model complexity, trade-offs between speed and performance, and limitations related to text diversity. In summary, the Transformer-based scene text detection and recognition method proposed in this paper has demonstrated outstanding performance in several aspects, and it holds great significance for advancing research and applications in the field of scene text processing.

VI. CONCLUSION

This paper investigated a Transformer-based method for scene text detection and recognition, and through experimental comparisons, it demonstrated the excellent performance and robustness of this approach in improving the precision and efficiency of both detection and recognition tasks. The paper also explored the positive effects and influences of end-to-end recognition, multi-scale fusion, and the Transformer framework on the results, as well as the effectiveness of this method in addressing the issue of noisy data inherent in images. By comparing with other methods, the paper concludes the following: End-to-end recognition reduces errors introduced between multiple independent steps, thereby enhancing the accuracy of both detection and recognition; Multi-scale fusion allows for text detection and recognition at different scales, improving model robustness and accuracy; The Transformer framework for feature extraction and classification enhances model efficiency. Experimental results demonstrate that the Transformer framework can learn relationships between text elements, thereby improving classification accuracy and efficiency. Compared to other methods, this approach significantly improves both accuracy and computational efficiency; Data augmentation techniques, such as image rotation, scaling, and cropping, enhance model robustness and accuracy. The experiments show that these techniques effectively address the issue of noisy data inherent in images. In future research, it is possible to explore additional techniques and methods, such as incorporating more semantic information, increasing model depth, and employing more sophisticated feature extraction methods, to further enhance the precision and efficiency of text detection and recognition. Additionally, in future work, considerations may include model lightweighting, multilingual support, and cross-domain applications to better meet real-world application needs and improve its performance, universality, and practicality.

REFERENCES

[1] P. Shivakumara, A. Banerjee, U. Pal, L. Nandanwar, T. Lu, and C.-L. Liu, "A new language-independent deep CNN for scene text detection and style transfer in social media images," *IEEE Trans. Image Process.*, vol. 32, pp. 3552–3566, 2023.

[2] E. Vidal, A. H. Toselli, A. Ríos-Vila, and J. Calvo-Zaragoza, "End-to-end page-level assessment of handwritten text recognition," *Pattern Recognit.*, vol. 142, Oct. 2023, Art. no. 109695.

[3] B. Ruzzante, L. D. Moro, M. Magarini, and P. Stano, "Synthetic cells extract semantic information from their environment," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 9, no. 1, pp. 23–27, Mar. 2023.

[4] W. Su, P. Miao, H. Dou, G. Wang, L. Qiao, Z. Li, and X. Li, "Language adaptive weight generation for multi-task visual grounding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 10857–10866.

[5] S. Lu, Y. Ding, M. Liu, Z. Yin, L. Yin, and W. Zheng, "Multiscale feature extraction and fusion of image and text in VQA," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, p. 54, Apr. 2023.

[6] F. Yuan, Z. Zhang, and Z. Fang, "An effective CNN and transformer complementary network for medical image segmentation," *Pattern Recognit.*, vol. 136, Apr. 2023, Art. no. 109228.

[7] J. Yi, Z. Shen, F. Chen, Y. Zhao, S. Xiao, and W. Zhou, "A lightweight multiscale feature fusion network for remote sensing object counting," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 3238185.

[8] J. Bharadiya, "Convolutional neural networks for image classification," *Int. J. Innov. Sci. Res. Technol.*, vol. 8, no. 5, pp. 673–677, 2023.

[9] S. Gujjeti and S. Radhika, "Analysis of various approaches for scene text detection and recognition," *J. Data Acquisition Process.*, vol. 38, no. 3, p. 1735, 2023.

[10] R. Li, S. Chen, F. Zhao, and X. Qiu, "Text detection model for historical documents using CNN and MSER," *J. Database Manage.*, vol. 34, no. 1, pp. 1–23, Apr. 2023.

[11] A. Roy, P. Shivakumara, U. Pal, H. Mokayed, and M. Liwicki, "Fourier feature-based cbam and vision transformer for text detection in drone images," in *Proc. Int. Conf. Document Anal. Recognit.* Springer, 2023, pp. 257–271.

[12] S. Zhang, A. Duan, and Y. Sun, "A text-detecting method based on improved CTPN," *J. Phys. Conf. Series*, vol. 2517, no. 1, 2023, Art. no. 012014.

[13] X. Shi, G. Peng, X. Shen, and C. Zhang, "TextFuse: Fusing deep scene text detection models for enhanced performance," *Multimedia Tools Appl.*, vol. 83, no. 8, pp. 22433–22454, Aug. 2023.

[14] X. Yan, H. Huang, Y. Jin, L. Chen, Z. Liang, and Z. Hao, "Neural architecture search via multi-hashing embedding and graph tensor networks for multilingual text classification," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 1, pp. 350–363, Feb. 2024.

[15] S.-X. Zhang, C. Yang, X. Zhu, and X.-C. Yin, "Arbitrary shape text detection via boundary transformer," *IEEE Trans. Multimedia*, pp. 1–14, 2023.

[16] J. Lin, Y. Yan, and H. Wang, "A dual-path transformer network for scene text detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[17] M. Krishnamoorthi, K. P. S. Ram, M. Sathyan, and T. Vasanth, "Improving optical character recognition(OCR) accuracy using multi-layer perceptron(MLP)," in *Proc. 7th Int. Conf. Trends Electron. Informat. (ICOEI)*, Apr. 2023, pp. 1642–1647.

[18] Y. Wu, L. Zhang, H. Li, Y. Zhang, and S. Wan, "Feature fusion pyramid network for end-to-end scene text detection," *ACM Trans. Asian Low-Resource Language Inf. Process.*, pp. 1–6, Jan. 2023.

[19] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "ABCNet: Real-time scene text spotting with adaptive bezier-curve network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9806–9815.

[20] H. Lu and H. Huo, "MSFRAN: Multi-scale feature fusion attention recognition network for text recognition in irregular scenes," *Int. Core J. Eng.*, vol. 9, no. 5, pp. 422–440, 2023.

[21] Y. Shi, X. Zhang, and N. Yu, "PL-transformer: A POS-aware and layer ensemble transformer for text classification," *Neural Comput. Appl.*, vol. 35, no. 2, pp. 1971–1982, Jan. 2023.
[22] A. Almutairi, B. Kang, and N. Fadhel, "The effectiveness of transformer-based models for bec attack detection," in *Proc. Int. Conf. Netw. Syst. Secur.* Cham, Switzerland: Springer, 2023, pp. 77–90.
[23] K. Manjari, M. Verma, G. Singal, and S. Namasudra, "QEST: Quantized and efficient scene text detector using deep learning," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 5, pp. 1–18, May 2023.
[24] S. Prabu and K. Joseph Abraham Sundar, "Enhanced attention-based encoder–decoder framework for text recognition," *Intell. Autom. Soft Comput.*, vol. 35, no. 2, pp. 2071–2086, 2023.
[25] G. Larbi, "Two-step text detection framework in natural scenes based on pseudo-zernike moments and CNN," *Multimedia Tools Appl.*, vol. 82, no. 7, pp. 10595–10616, Mar. 2023.
[26] S. Tarride, M. Maarand, M. Boillet, J. McGrath, E. Capel, H. Vézina, and C. Kermorvant, "Large-scale genealogical information extraction from handwritten Quebec parish records," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 26, no. 3, pp. 255–272, Sep. 2023.
[27] R. Bhatt, A. Rai, S. Chanda, and N. C. Krishnan, "Pho(SC)-CTC—A hybrid approach towards zero-shot word image recognition," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 26, no. 1, pp. 51–63, Mar. 2023.
[28] Z. Luo, X. Zeng, Z. Bao, and M. Xu, "Deep learning-based strategy for macromolecules classification with imbalanced data from cellular electron cryotomography," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
[29] D. K. Vishwakarma, P. Meel, A. Yadav, and K. Singh, "A framework of fake news detection on web platform using ConvNet," *Social Netw. Anal. Mining*, vol. 13, no. 1, p. 24, Jan. 2023.
[30] P. Naveen, *End-to-end Training of VAE-GAN Network for Text Detection*, 2023.
[31] H. Chen, Y. Qiu, M. Jiang, J. Lin, and P. Chen, "Kernel-mask knowledge distillation for efficient and accurate arbitrary-shaped text detection," *Complex Intell. Syst.*, vol. 10, no. 1, pp. 75–86, Feb. 2024.
[32] Z. Luo, H. Xu, and F. Chen, "Audio sentiment analysis by heterogeneous signal features learned from utterance-based parallel neural network," in *Proc. AffCon@ (AAAI)*, 2019, pp. 80–87.
[33] H. Wang, H. Shan, Y. Song, Y. Meng, and M. Wu, "Engineering drawing text detection via better feature fusion," in *Proc. Int. Conf. Ind., Eng. Appl. Applied Intell. Syst.* Cham, Switzerland: Springer, 2023, pp. 265–270.
[34] S. He, "Fabrication and control of porous structures via layer-by-layer assembly on PAH/PAA polyelectrolyte coatings," *Biomed. J. Sci. Tech. Res.*, vol. 51, no. 5, pp. 1–4, Jul. 2023.
[35] A. R. Rashtehroudi, A. Akoushideh, and A. Shahbahrami, "PESTD: A large-scale persian-english scene text dataset," *Multimedia Tools Appl.*, vol. 82, no. 22, pp. 34793–34808, Sep. 2023.
[36] J. Zheng, W. Li, J. Hong, L. Petersson, and N. Barnes, "Towards open-set object detection and discovery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3960–3969.
[37] S. He, "Hybrid dual-functional Au-on-Ag nanostructure for monitor. Au-catalyzed reactions in situ by surface-enhanced Raman scattering," Ph.D. thesis, Stevens Inst. Technol., 2022.
[38] W. Dai, C. Mou, J. Wu, and X. Ye, "Diabetic retinopathy detection with enhanced vision transformers: The twins-PCPVT solution," in *Proc. IEEE 3rd Int. Conf. Electron. Technol., Commun. Inf. (ICETCI)*, May 2023, pp. 403–407.
[39] F. Chen, Z. Luo, Y. Xu, and D. Ke, "Complementary fusion of multi-features and multi-modalities in sentiment analysis," 2019, *arXiv:1904.08138*.
[40] Z. Luo, "Knowledge-guided aspect-based summarization," in *Proc. Int. Conf. Commun., Comput. Artif. Intell. (CCCAI)*, Jun. 2023, pp. 17–22.

TIANYU GENG is currently pursuing the bachelor's degree with the College of Computer and Information Engineering (College of Artificial Intelligence), Nanjing Tech University. His research interests include machine learning, deep learning, and computer vision.