

Article

Scene Text Recognition That Eliminates Background and Character Noise Interference

Shancheng Tang , Yaoqian Cao *, Shaojun Liang, Zicheng Jin  and Kun Lai

College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China; tangshancheng@xust.edu.cn (S.T.); 20207040019@stu.xust.edu.cn (S.L.); 21207035005@stu.xust.edu.cn (Z.J.); 21307223009@stu.xust.edu.cn (K.L.)

* Correspondence: 22207223074@stu.xust.edu.cn

Abstract: In natural photographs, complex background noise and character noise frequently interfere with scene text identification. To solve the aforementioned concerns, this paper proposes a novel scene character identification model that eliminates noise from both the backdrop and the character (ENBC). The model is divided into three pieces. To begin, the high-level character feature extraction module uses ASPP dilated convolution with varying expansion rates to obtain features at various scales, thereby expanding the receptive field to capture the character feature area more effectively, eliminating noise interference from the character itself, and improving the character shape features. Second, the multi-level character feature fusion module merged the high-level character feature information after upsampling with the low-level character feature information in the backbone network, separated the foreground characters from background interference, removed background noise, and output the resulting image. Third, the recognition enhancement module enhances character context modeling by considering both forward (left-to-right) and backward (right-to-left) information from the text sequence. The experimental results show that the model can effectively minimize background and character noise interference, boosting recognition accuracy by at least 4.2% on the synthetic scene dataset. When compared to other popular techniques on the IIIT5K, ICDAR-2015, ICDAR-2003, and CUTE80 public datasets, recognition accuracy improves by an average of 6.97%.



Academic Editor:
Douglas O'Shaughnessy

Received: 5 January 2025

Revised: 13 March 2025

Accepted: 19 March 2025

Published: 24 March 2025

Citation: Tang, S.; Cao, Y.; Liang, S.; Jin, Z.; Lai, K. Scene Text Recognition That Eliminates Background and Character Noise Interference. *Appl. Sci.* **2025**, *15*, 3545. <https://doi.org/10.3390/app15073545>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: scene text recognition; deep learning; background noise; character's own noise; enhanced character features

1. Introduction

Scene text recognition (STR) is an emerging study topic in computer vision, with applications including robotics process automation, picture retrieval, autonomous driving, real-time translation, and more. STR is the mechanical extraction of text information from pictures or movies in realistic situations, and it is more complex than optical character recognition (OCR) operations [1,2]. This is because the text in images of natural scenes has multiple languages, rich font styles, diverse colors, variable scales, and free arrangements, among other issues. Meanwhile, the backgrounds in natural scene images are complex and variable with serious noise interference, and some background objects and texts are actually difficult to distinguish.

As a response to the aforementioned challenges, a variety of text recognition technologies are presently presented, which may be classed as classical or deep learning approaches based on the manner of feature extraction. In well-known conventional approaches, character pictures are often represented using manual characteristics [3–5]. To extract character

picture feature information, approaches such as co-occurrence histogram of oriented gradients (co-HOG), histogram of oriented gradients (HOG), and convolutional co-occurrence histogram of oriented gradients (ConvCo-HOG) were utilized. For example, Su et al. [5] investigated an effective extraction approach by converting picture information into consecutive column vectors using HOG algorithms. Casey et al. [6] recognized printed text using iterative and heuristic techniques; Zhou et al. [7] utilized the K-mean approach to coarsely classify the characters and retrieved the character feature information in eight directions and then finely classed the character feature information using the improved minimum quality discriminant function (MQDF) classifier; Qu et al. [8] presented the adaptive discriminative localization alignment (ADLA) approach for similar character detection based on character similarity to address the annoyance created by similar characters in pictures. However, traditional methods of recognizing text are limited to the artificial design of feature expression ability and processing complexity. It is challenging to attain higher identification accuracy in increasingly complicated visual scenarios, such as text images with significant angular distortion and blurring.

As deep learning technology advances rapidly in the field of text recognition, text feature extraction methods have gradually evolved from manual design to automatic extraction using deep neural network models, resulting in optimized recognition performance and significantly improved the accuracy. Deep learning-based text recognition systems typically involve four major steps: preprocessing, feature extraction, sequence modeling, and prediction. During the preprocessing stage, the input picture is denoised, greyscaled, binarized, and scaled to increase recognition accuracy and efficiency. Then, in the feature extraction step, from the preprocessed image, a deep convolutional neural network (CNN) is applied to extract local visual information, which our technique primarily improves upon to tackle the problem effectively from the root. The long-range relationships between distinct characteristics are explored further during the sequence modeling step. Finally, in the prediction step, prediction results for text occurrences are generated [9]. In this way, STR systems may be classified into two types depending on the application dimension of the attention mechanism: attention that is both one- and two-dimensional. Figure 1 depicts the picture character recognition method from both one-dimensional and two-dimensional perspectives.

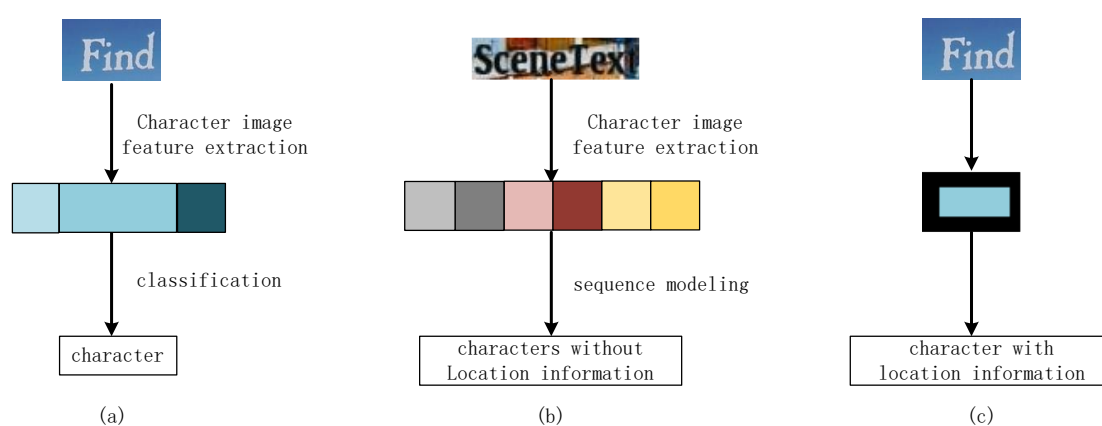


Figure 1. Character recognition technique. (a) Character image recognition in one dimension; (b) character image recognition with sequence prediction in a 1D perspective; (c) a character image recognition method based on semantic segmentation.

1.1. A Single-Dimensional Attention-Based Method for Scene Text Recognition

The majority of existing recognition systems employ Convolutional Neural Networks (CNNs) [10] to automatically extract various degrees of character feature information from

pictures [11–13]. In general, these approaches reduce character pictures to one-dimensional characteristics. For example, ref. [12] proposed convolutional recurrent neural network (CRNN) for sequence annotation by combining CNN and recurrent neural network (RNN), in which they deal with character sequences of variable lengths while reducing reliance on segmentation. He et al. [13] suggested a technique to analyze character sequences that combines connected temporal classification (CTC) [14] with long short-term memory (LSTM) to address the issue of irregular spacing between characters as well as the difficulty of character alignment. Figure 1a,b demonstrates two instances of one-dimensional character and text recognition. Although these approaches are effective, they compress a character's attributes into a one-dimensional shape without taking into account the character's structural information, creating background noise interference [15].

1.2. A Two-Dimensional Attention-Based Method for Scene Text Recognition

Overall, this text recognition approach converts the original one-dimensional predictive model to a two-dimensional (2D) version by creating a 2D decoder based on the attention mechanism [15–17], allowing the sequence model to use all geometric and character structure information for character feature alignment. Furthermore, there is an increasing tendency of merging visual and verbal data to improve recognition. For instance, Nguyen et al. [18] used dictionaries as linguistic information to improve initial recognition results, ref. [19] proposed an iterative linguistic structure to correct initial recognition results, and Tounsi [20] et al. proposed a hybrid bag of features-stacked autoencoder-hashing attention model (BoF-SAE-HAM) design for recognizing words in natural scenes. In addition, investigating improved character representations is a promising avenue for scene word recognition. Ref. [21] sought to create basic representations on undirected graphs and utilize graph convolutional network (GCN) to improve character recognition. In the last several years, deep learning techniques based on semantic segmentation have steadily been used to STR. Ref. [15] created a fully convolutional network based on the character attention mechanism to recognize scene text and forecast characters' exact locations. Although the approaches mentioned above have had some success, the current STR methods are only relevant to standard English typefaces and a few categories of numerical identification, and the recognition accuracy is low. Combining the aforementioned issues, this study provides a scene text recognition approach to identify large-scale characters using pixel-by-pixel prediction from a two-dimensional viewpoint, and applies it to several irregular English and numeric characters, as illustrated in Figure 1c.

Background noise and character noise are frequently regarded as worthless and impeding information for text recognition. To address the aforementioned issues, an STR model called the eliminating noise from background and character network (ENBC) is proposed. The model is made up of three modules: high-level character feature extraction, multi-level character feature fusion, and recognition improvement. In the feature encoding stage, the scene image is convolved with ASPP voids in the high-level character feature extraction module to capture the character feature information and eliminate the noise interference of the characters. In the decoding and reconstruction stage, the image is fused with different levels of character feature information through the multilevel character feature fusion module to eliminate the background noise, and then the reconstructed image is The reconstructed picture is then merged with cross entropy and an auxiliary loss function to indefinitely approximate the label image; finally, the reconstructed picture is upgraded by the recognition enhancement module to increase character context modelling capacity and text recognition accuracy, and the method's reliability and efficacy are tested using tests. This publication contributes the following:

1. A new scene text recognition method is proposed to eliminate background and character noise interference. The method uses the lightweight MobileNetV2 as the base network and employs ASPP dilated convolutions with different dilation rates to capture features at different scales, thereby expanding the receptive field. This enables more effective capturing of character feature regions, eliminating character noise interference, and enhancing character shape features. By combining the high-level character feature information obtained through sampling with the low-level character information in the backbone network, the foreground characters and background interference are separated, and background noise is eliminated.

2. A recognition improvement module is designed to learn text basic representation. The context modeling capacity is improved by taking into account both forward (left-to-right) and backward (right-to-left) information of text sequences, resulting in increased text recognition accuracy and resilience.

2. Related Work

2.1. Text Recognition of Scenes Using Deep Learning

General optical character recognition approaches are unable to effectively address the varied backgrounds, different character sets, uneven lighting, and poor resolution that characterize scene character recognition. The early approaches in STR mainly depended on hand-crafted features to overcome the difficulties. When extracting features from photos with intricate backgrounds and characters, the HOG approach is employed [22,23] presented two new feature descriptors, ConvCo-HOG and Co-HOG, which are based on the HOG approach. In order to conduct word recognition without character segmentation, Su et al. [24] employed HOG to transform word pictures into sequence features, which they then fed into LSTM and CTC. Even while these techniques work well, they have drawbacks, such as the fact that they are often intended for particular scenarios.

In recent years, to address various objectives, researchers have attempted to enhance feature extraction networks in recent years from several angles. Recurrent convolutional neural network [25] proved well in picture classification tests, and Wang et al. [26] used this technique to extract features from a natural scene text recognition network. In order to further optimize the information modeling process, they also devised a gating mechanism. Albahli et al. [27] presented a novel method for automatically recognizing and categorizing handwritten digits by utilizing the DenseNet-41 framework in the feature extraction stage in conjunction with the faster kernelized convolutional neural network (Faster-KCNN) deep learning algorithm. A continuous convolutional activation (CCA) technique was developed by Zhang et al. [28] that integrates both high-level and low-level patterns into a single feature vector. Usually, these algorithms generally transform a 2D image into a 1D feature vector. As a result, they produce noise in the background and have no ability to localize character positions. Therefore, in order to get the network to concentrate on retrieving foreground text information, several researchers have coupled convolutional processes with attention mechanisms [29]. Liu et al. [30] suggested using a binary strategy to enhance the feature extraction convolutional network to comply with the requirement for real-time recognition of text from natural scenes. This approach can minimize memory occupation and speed up inference. To achieve optimal performance, the feature extraction network must undergo extensive and time-consuming parameter tweaking on various datasets.

Scene texts have a wide and complicated structure. As a result, several researchers have created generative tasks [31–33] to identify distinctive characteristics. Wang et al. [34] devised a way to learn the canonical forms of glyphs in many popular font styles using a generative adversarial network (GAN). The framework is made up of four networks: feature extraction, character categorization, character production, and character discrimination.

Ref. [35] suggested multi-task collaborative generative adversarial network (MtC-GAN), a network architecture for producing actual data, to replicate real-world datasets and increase STR accuracy. However, both [34,35] rely on uncontrolled font renderers to create standard print characters, which is extremely limited for recognition.

2.2. Scene Text Recognition Method Based on Image Segmentation

Segmentation networks must have excellent modeling capabilities in order to make correct judgments regarding text pixels, background pixels, and the transition zone between them; otherwise, mis-segmentation such as over-segmentation, missing, blurring, and so on may occur. The aforementioned problems are somewhat alleviated by the scene text segmentation methods now in use, but there is still room for improvement. Spatial and multi-scale attention network (SMANet) [36] employs a pooling technique to extract multiple scale text characteristics; however, it loses spatial location information for scene text and is unable to maintain feature map resolution. Multi-granularity network (MGNet) [37] uses an instruction under partial supervision technique with text polygonal box mask labels to offer a priori knowledge for pixel-level text segmentation, which aids in determining text position; nevertheless, it lacks a network architecture for natural scene text features. Deep convolutional neural network with transfer learning (Deep-CNN TL) [38] combines deep convolutional neural network (DCNN) with TL to enhance text boundary detection; however, the approach still has a lot of space for improvement when dealing with excessively noisy scene photos. Text recognition network (TexRNet) [39] employs the same strategy in the generic segmentation backbone network [40–43] after developing an improved network that leverages cosine similarity to repair misclassified text pixels with unusual textures and styles. TexRNet is customized based on the scene text properties, which can lead to improved segmentation results. The potential for improving generic segmentation networks in scene text segmentation tasks is proven, although the discriminator in this technique must be trained using complicated character-level annotations.

Semantic segmentation, in contrast to ordinary segmentation, categorizes each pixel in a picture. Semantic segmentation [40,44–47] is a common computer vision problem for identifying the category, position, and form of items in a picture. The fully convolutional network (FCN) [48] framework design has been used to extend most semantic slicing techniques, including semantic image segmentation with deep convolutional Nets, atrous convolution, and fully connected CRFs (DeepLab) [49], convolutional networks for biomedical image segmentation (U-Net) [50], pyramid scene parsing network (PSPNet) [51], and others. Refs. [15,52] successfully employed semantic segmentation models to fulfill the STR challenge. However, they can only detect text in basic scenarios. Liao et al. [52] also introduced a segment anything model (SAM) to the mask an end-to-end trainable scene text spotting system (TextSpotter) [52] model to address the multilingual end-to-end recognition challenge. However, when detecting text in complicated contexts, the character segmentation branch is muted, and sequence prediction is handled only by the SAM [52], resulting in some recognition restrictions. The work in this study is based on a traditional semantic segmentation framework and the design of an ENBC network to extract features for pixel-by-pixel classification, with the purpose of reducing background noise with the interference of the character's own noise, enhancing the character's shape features, and improving text recognition accuracy.

3. Proposed Framework

Figure 2 depicts the paper's model structure, which is divided into three parts: pre-processing, the ENBC network, and scene text recognition. The data synthesis tool generates the dataset needed for the ENBC network in Figure 2a. Figure 2b consists of the encoder and

decoder. The encoder adopts the relatively lightweight backbone network MobileNetV2, and expands the receptive field by obtaining features of different scales using the ASPP null convolution with different expansion rates, so as to get the contextual character feature information with a larger receptive field and eliminate the character's own noise, while the decoding stage fuses different levels of character feature information by Eliminate the character's own noise, while the decoding step eliminates background noise by fusing multiple levels of character feature information using the multilayer character feature fusion module. A multilayer character feature fusion module, a high-level character feature extraction module, and a recognition improvement module make up Figure 2c. The multilevel character feature fusion module eliminates background noise and produces a reconstructed image; the high-level semantic feature extraction module broadens the receptive field through ASPP null convolution to obtain character feature information with a larger receptive field; and the recognition enhancement module improves contextual modeling capability by taking into account both the forward (left-to-right) and backward (right-to-left) information of the text sequences simultaneously, improving text recognition accuracy and robustness.

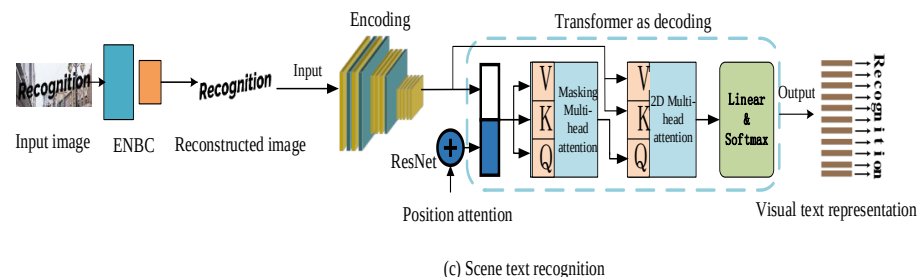
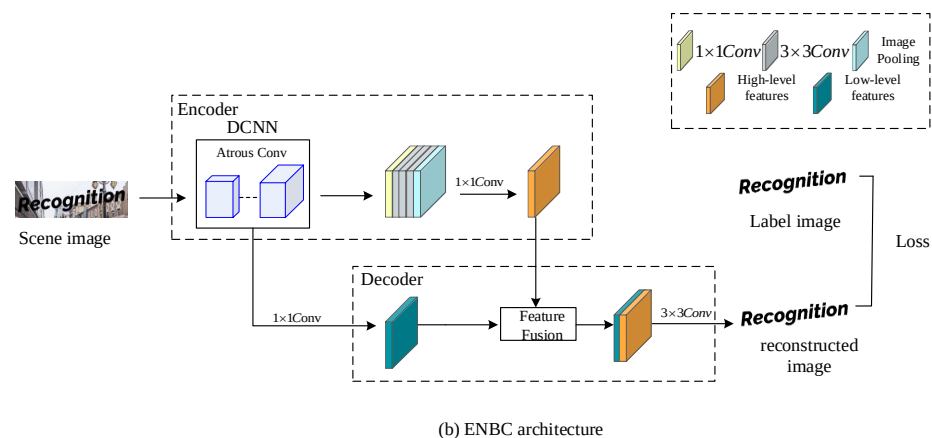
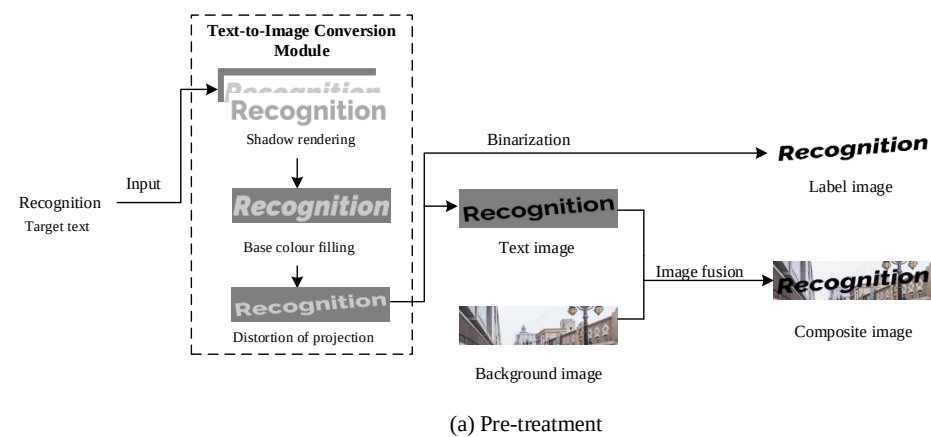


Figure 2. Scene text recognition framework to eliminate background and character noise interference.

3.1. Pre-Processing

This study generates the dataset needed for the ENBC network using the Baidu Flying Paddle Text Recognition Data Generator tool. The bulk of text in natural contexts is limited to a small number of computer-generated fonts and is created by physical rendering and imaging processes that are not governed by computer algorithms [53]. The Text Recognition Data Generator is a synthetic word data generator that mimics text picture scene distribution.

Three distinct picture layers—a background image layer, a foreground image layer, and an optional border/shadow image layer—are created throughout the dataset production process and are represented by images with alpha channels. Figure 3 shows the creation procedure as well as a few examples of the synthetic data. Following font rendering, image layer creation and shading, projection distortion application, and image blending, Figure 3a illustrates the text generation process. Figure 3b depicts some drawn-at-random data generated by the Synthesized Text Tool. The synthetic data-generating procedure is as follows:

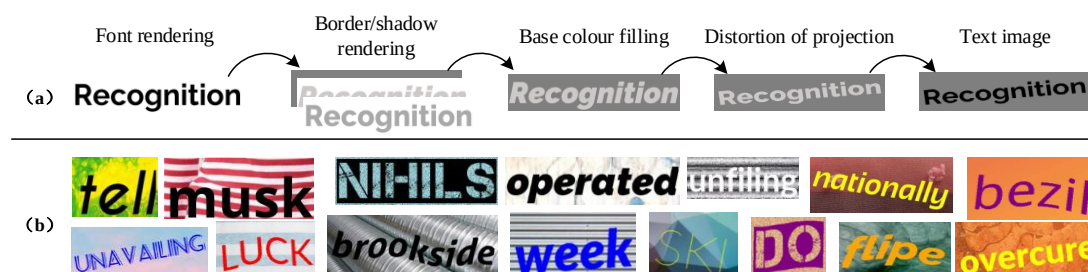


Figure 3. Process of creating synthetic data. (a) The process of creating text; (b) a little data picked at random.

1. Font rendering—Choose a typeface at random from the Google Fonts download's collection of more than 1400 fonts. Word counts, margins, underlining, and other characteristics are randomly distributed using a distribution that may be set in any way. The word is produced using either horizontal bottom text lines or random curves on the alpha channel of the foreground picture layer.

2. Border/shadow rendering—The foreground rendering allows for the insertion of a border, start border, or shadow with any width.

3. Base color filling—A distinct, homogeneous color is sampled from clusters on the original image to fill each of the three image layers. The RGB elements of every picture in the training dataset are divided into three groups to form these clusters.

4. Distortion of projection—To imitate the 3D environment, random complete projection modifications are used on the foreground and border/shadow picture layers.

5. Text image—Randomly selected picture segments from the SVT and ICDAR 2003 training datasets are mixed into each image layer. A stochastic process determines the quantity of blending and alpha blending modes (normal, add, multiply, burn, max, etc.), producing a diverse array of textures and compositions. To create a single text picture, the three image layers are also randomly mixed together.

6. Noise—The image is subjected to JPEG compression artifacts, blurring, resampling noise, and Gaussian noise.

Through the use of several random distributions that mimic actual text pictures of scenes, this approach creates a vast array of synthetic data samples. Real-world data are replaced with synthetic data, and tags are created as needed from a corpus or dictionary. Deep learning methods that have a high demand for data can be used to train richer

whole-word-based models by providing an order of magnitude bigger training dataset than ever before.

3.2. The ENBC System

Figure 2b displays the ENBC network structure. Prior to being normalized to $[-1, 1]$, the image must first be scaled and standardized to a size of $[150, 500]$. To ensure that it expands the receptive field and more successfully captures the character's feature area while removing noise interference from the character itself, the encoder employs ASPP dilated convolution with varying expansion rates during the feature encoding stage. Upon up-sampling the high-level character feature information and combining it with the low-level character feature information in the backbone network to remove background noise, the decoder produces the reconstructed picture during the decoding and reconstruction step.

3.2.1. Encoder

Text recognition relies heavily on character context information. However, the size, form, and position of writing in natural settings vary widely. Previous techniques often integrate contextual information via consecutive pooling or other downsampling layers, resulting in a loss of resolution. Furthermore, the idea of receptive fields in convolutional neural networks is quite significant. When the receptive field is too tiny in comparison to the segmented picture, it captures the image's local information, and if the global information is not present, it causes various levels of segmentation problems. As a result, the size of the receptive field directly influences how much image feature information is captured, affecting reconstruction outcomes. Traditional recognition techniques with inadequately large receptive fields will result in decreased recognition accuracy for longer or bigger texts. Instead, the ENBC network captures information at various sizes using atrous convolution and atrous spatial pyramid pooling (ASPP).

Dilated convolutions broaden the receptive field by introducing gaps (dilations) between the convolution kernel's components, enhancing information characteristics between characters. At the same time, this process must ensure that both the processing cost and parameter count remain constant. This empowers the system to collect more character context information while still keeping high-resolution features. Assuming $k \times k$ is the size of the convolution kernel, $H \times W$ is the dilation rate, and $y[i, j]$ is the size of the input feature map, the result of the convolution operation may be stated as:

$$y[i, j] = \sum_{m=1}^k \sum_{n=1}^k x[i + r \times (m - 1), j + r \times (n - 1)] \cdot w[m, n] \quad (1)$$

Among them, w is the weight of the convolution kernel; x is the input feature map; y is the output feature map; and r is the expansion rate, which governs the spacing of elements in the convolution kernel and defines the size of the receptive field. As a result, null convolution is utilized to produce features with the same size but distinct receptive fields. The emergence of null convolution [54] can solve the contradiction between resolution and receptive field in the pooling layer, and the resulting feature mapping can have the same size as the input image while increasing the receptive field without increasing the number of convolution kernels, which is increasingly being used by researchers. The idea of dilatation rate is introduced in null convolution, as seen schematically in Figure 4 below. Figure 4a depicts a 3×3 kernel and a 1 expansion rate, also known as conventional convolution. Figure 4b depicts a 3×3 kernel and a 2 expansion rate. For a 7×7 picture block, just 9 points and 3×3 convolution kernels are utilized for convolution, leaving the other points unaffected. Figure 4c illustrates a 3×3 kernel and a 4 expansion rate, resulting in a 15×15 receptive field. Dilated convolution can totally replace the pooling procedure,

increasing the receptive field without affecting the picture quality, resulting in an output with a wide range of character image feature information.

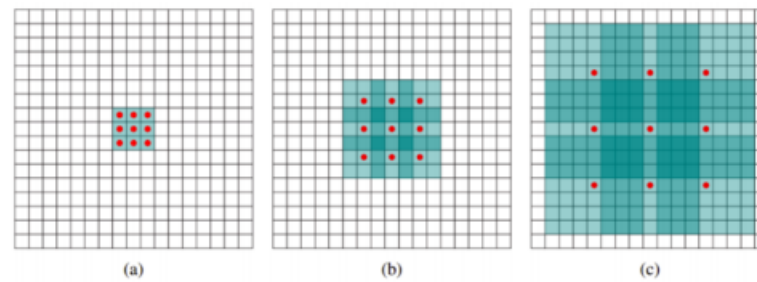


Figure 4. Schematic diagram of the cavity convolution. (a) Ordinary convolution (conv 3×3 , rate1); (b) null convolution (conv 3×3 , rate2); (c) null convolution (conv 3×3 , rate4).

The ASPP module in the ENBC network extends the output feature mapping to cover larger scales of sensory field sizes by applying several null convolution layers in parallel, each with a distinct expansion rate, as illustrated in Figure 5. The ASPP has five branches, including a 1×1 convolution, three cavity convolutions with variable expansion rates, and a global average pooling branch. Each branch represents a different scale, allowing for the acquisition of higher-level character picture information.

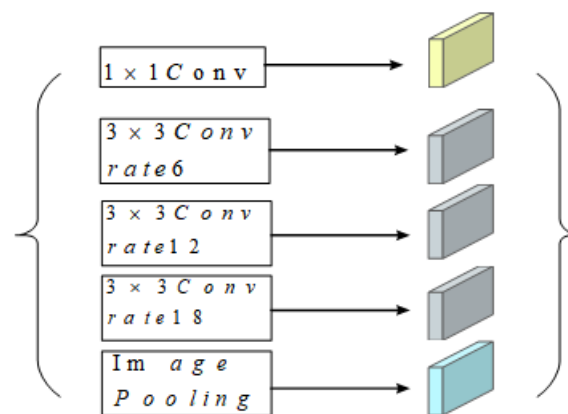


Figure 5. ASPP structure.

The output of ASPP can be expressed as:

$$y_{ASPP} = \text{Concat}(\text{Conv}_{1 \times 1}(x), \text{Conv}_{3 \times 3}(x, r_1), \text{Conv}_{3 \times 3}(x, r_2), \text{Conv}_{3 \times 3}(x, r_3), \text{Pooling}(x)) \quad (2)$$

These include the following: r_1 , r_2 and r_3 are distinct expansion rates; x , the input feature map; $1 \times 1 \text{Conv}$, the convolution; *Pooling*, the global average pooling operation; and *Concat*, the feature splicing operation.

3.2.2. Analysis of Deep Convolutional Neural Network Models

Through layered convolutional and pooling layers, DCNN gradually recovers low-level and high-level aspects of the picture, such as texture and geometric information, in deep learning-based word recognition algorithms for scene images. The foundational DCNN frameworks VGGNet, ResNet, Inception, and MobileNetV2 are often utilized in the picture domain. A lightweight network among them is called MobileNetV2 [55], which was introduced by Google in 2017. Its main feature is the substitution of depth-separable convolution for the conventional convolution. Depth-separable convolution can decrease the calculation to $1/8$ – $1/9$ of the original by splitting the normal convolution into 1 depth convolution and 1 point-by-point convolution (also known as 1×1 convolution).

Figure 6 displays the depth-separable convolution + BN + ReLU network versus the regular convolution + BN + ReLU network.

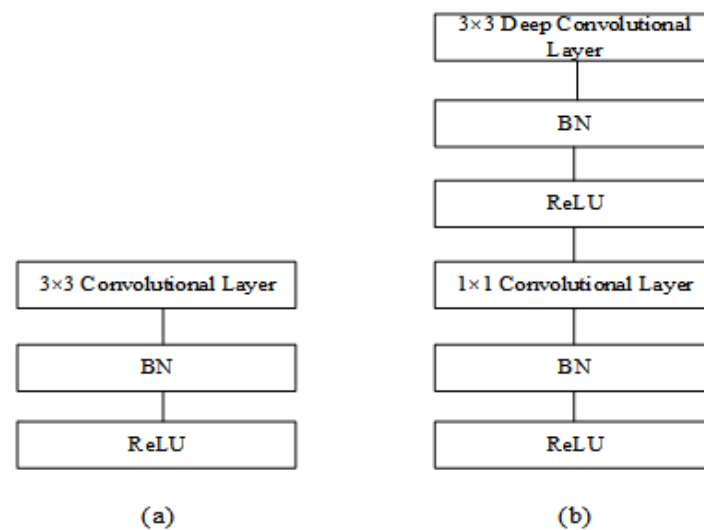


Figure 6. (a) Standard convolution + BN + ReLU network; (b) depthwise separable convolution + BN + ReLU network.

In this research, the MobileNetV2 architecture is used by the ENBC network, which is built on a simplified design that builds a lightweight deep neural network with high-performing trade-offs for model accuracy using highly separable convolutions. Figure 7 depicts the MobileNetV2 [56] structure, which adopts a more high-performing residual structure, suggests an inverse residual module, and converts the MobileNet v1 module's last ReLU6 layer to a linear layer.

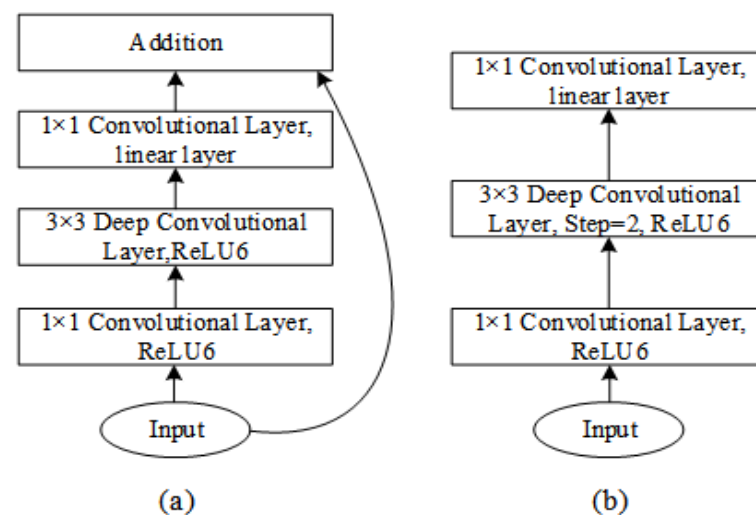


Figure 7. MobileNetV2 module. (a) Module with step 1; (b) module with step 2.

3.2.3. Decoder

Decoders are frequently employed in ENBC networks to increase image reconstruction resolution, particularly for recovering high-resolution picture features. To produce the final reconstruction result, the decoder combines the encoder's high-level character semantic data with the low-level spatial detail characteristics. In the network architecture suggested

in this study, the input to the decoder is the DCNN output $y_{LowLevel}$, and the ASPP module output y_{ASPP} . The decoder's feature fusion formula is as follows:

$$Y_{Decoder} = Concat(y_{ASPP}, y_{LowLevel}) \quad (3)$$

Among these, y_{ASPP} has picture spatial information with a significant number of channels. To limit the number of channels to 256 for further processes, a 1×1 convolution is performed first. To get more accurate picture reconstruction results, the decoder must integrate the low-level information $y_{LowLevel}$ with high-level information y_{ASPP} retrieved from the network. High-level feature maps from the other decoder input are utilized to offer more global and abstract character semantic information, while low-level feature maps from the DCNN are used to provide edge and detail information. The two feature maps are first spliced in the channel dimension; next, the channel dimension is compressed using a 3×3 convolution; and lastly, an up-sampling operation is performed using bilinear interpolation to reduce the feature map resolution to the input image size, in order to obtain the final reconstruction result of the network. This process aims to better fuse the $y_{LowLevel}$ and the y_{ASPP} .

3.2.4. Loss of Model

The choice and configuration of the loss function in deep learning are essential to the model's performance and training. By minimizing the loss function, the model may progressively learn to make predictions that are increasingly accurate. The loss function is used to assess the difference between the model's predictions and the actual labels. There are two components to the ENBC network's loss function.

(1) The network's principal purpose is cross-entropy loss, which is determined by calculating the cross-entropy error between each pixel point's actual label and the category's projected probability. This may be represented as follows:

$$L_1 = -\frac{1}{n} (y_i (\ln \alpha) + (1 - y_i) \ln(1 - \alpha)) \quad (4)$$

$$\begin{aligned} \frac{\partial L_1}{\partial w_j} &= \frac{1}{n} \sum_x x_j (\sigma(z) - y), \\ \frac{\partial L_1}{\partial b} &= \frac{1}{n} \sum_x (\sigma(z) - y) \end{aligned} \quad (5)$$

In the formula:

$$z = wx + b \quad (6)$$

Among them, L_1 represents the loss value, x the sample, y the actual value, or the predicted value, α the output value, and n the total number of samples. The gap between the output value and the actual predicted value is then represented by parameter $\sigma(z) - y$, and a bigger mistake corresponds to a larger gradient, which causes the parameters w and b to be modified more quickly, potentially increasing the training speed. The network may be optimized by continuously adjusting the size of these parameters throughout the actual training phase to lower the loss value.

(2) An additional loss function is employed by the network in addition to the cross-entropy loss. Its moderating element, which is based on the cross-entropy loss, is included to lessen the overfitting of the model to the samples. It is expressed as follows:

$$L(p, t) = -\sum w[t] (1 - p[t]) \gamma \log(p[t]) \quad (7)$$

The weighting factor (w), the actual label (t), the category probability (p), and the adjustable parameter (γ), which controls the factor's intensity, are all part of the equation above. Consequently, the model's overall loss is:

$$Loss = L_1 + L(p, t) \quad (8)$$

3.3. Scene Text Recognition Model

The structure of the scene text recognition model, which removes character and background noise interference, is depicted in Figure 8 below. By combining a multilayer character feature fusion module with a high-level character feature extraction module, the ENBC network reconstructs the scene picture while improving the character shape features and removing background and character noise. After the image has been rebuilt, it is fed through the recognition enhancement module, which uses the ResNet network to handle noise and distortion in the input data and successfully extract character image features at various scales. The transformer is then used as a decoder to generate a rough text prediction.

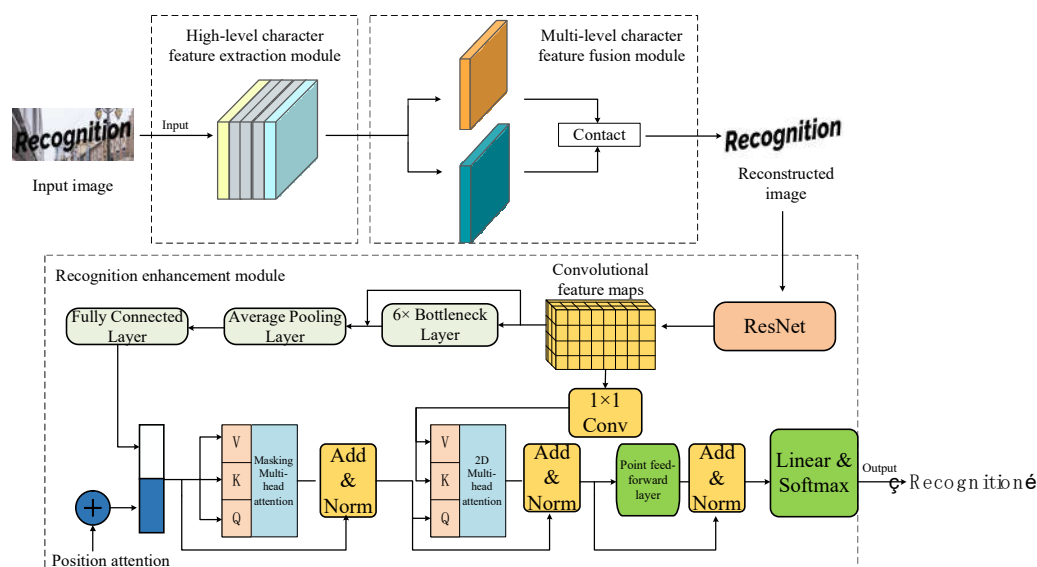


Figure 8. Structure of scene text recognition model eliminating background and character's own noise interference.

In order to extract character image characteristics at various scales, the rebuilt picture is sent into the ResNet network. The ResNet feature map is split into two branches: an overall representation extractor with six bottlenecks, average pooling, and a holistic representation extractor with a fully connected layer that handles noise and distortion in the input data, and a 1×1 convolutional layer that converts the 2D feature map's dimension from to [1024] for input to the 2D attention. The output channel and kernel size are indicated by the letters "Conv", which stand for convolutional layer. The convolutional layer's step size and padding are both set to "1".

After that, the reconstructed picture is sent through the mask self-attention mechanism, 2D attention module, and point-by-point feed-forward layer decoding section using transformer as the decoder. The point-by-point feedforward layer is applied to each decoding location, and the mask self-attention mechanism is utilized to simulate the relationships between various characters inside the output word. Layer normalization is the step that comes after additive operations and residual concatenation in each of the aforementioned three layers.

According to Figure 8, the recognition improvement module embeds the previously created characters at each decoding step into a $d/2$ -dimensional vector, which is then further summed with the encoding of the current location. This process is known as positional encoding and embedding. Here is the formula for position encoding.

$$PT(t, i) = \begin{cases} \sin(t/10,000^{i/(d/2)}) \\ \cos(t/10,000^{(i-1)/(d/2)}) \end{cases} \quad (9)$$

where the position is represented by t , the dimension by $i \in \{1, \dots, (d/2)\}$, and the position encoding by PT . During training, they are combined into an overall representation in which real characters are concurrently inserted and right-shifted.

The decoder's two-dimensional attention mechanism and the masked self-attention mechanism both rely on the multi-headed dot product attention formulation [57]. A query $q \in R^d$ and a set of d dimensional vectors of key-value pairs $\{(x_i, y_i)\}_{i=1,2,\dots,M}$ (where M is the number of key-value pairs) are given to scaled dot product attention, which then computes a weighted sum of the output values. Each value's weight is determined by scaling the dot product of the query and its corresponding key. This is the formulation for the scaled dot product mentioning.

$$Attention(Q, K, V) = \sum_{i=1}^M \alpha_i y_i \in R^d \quad (10)$$

$$\alpha = softmax(\frac{\langle q, x_1 \rangle}{\sqrt{d}}, \frac{\langle q, x_2 \rangle}{\sqrt{d}}, \dots, \frac{\langle q, x_M \rangle}{\sqrt{d}}) \quad (11)$$

where the attention weight is α , and among them, $X = [x_1, x_2, \dots, x_M]$ and $Y = [y_1, y_2, \dots, y_M]$. We can get the formula if there is a set of queries $Q = [q_1, q_2, \dots, q_{M'}]$ (there are M' inquiries total).

$$Attention(Q, K, V) = [a_1, a_2, \dots, a_{M'}] \in R^{d \times M'} \quad (12)$$

$$a_i = Attention(q_i, K, V) \quad (13)$$

where the query, key, and value are the same across several decoding places, the relationships between them are modeled using the mask self-awareness layer. The decoding sequence in this instance has a length of $M = M'$. Every position is covered by a mask that keeps it from focusing on the position that comes after it.

The key and value are the encoder's 2D output features, and the query is generated by the 2D attention layer's masked self-attention. In this instance, the decoded sequence's length is $M = 6 \times 20, M'$. It serves as the primary link between the encoder and the decoder and enables the recording of the encoder output's 2D location at each decoding point.

Each location of the 2D attention layer output is subjected to a basic feed-forward network in the point-by-point feed-forward layer. This layer's parameters are shared at all points and comprises two linear transformations of dimension d' as well as a ReLU nonlinear between.

Ultimately, to decode the visual text representation and output the recognition results, a fully linked layer with a softmax function is employed.

4. Experimental Results and Analyses

First, train and test the ENBC network, then build the model in this paper based on the ENBC network. The model will be tested on both synthetic and real-world datasets, and experiments will be conducted on multiple publicly available real-world scenario

datasets, comparing it with some current classical models and the latest models to validate the effectiveness of the proposed model.

4.1. Experimental Datasets

The datasets utilized in the trials are split into a synthetic scene dataset and an actual scene dataset in order to address the challenges of low levels of data with annotations and high costs associated with manual annotation in natural settings, as well as to confirm the model's generalization.

(1) Synthetic scene dataset

Preprocessing yields the synthetic scene dataset text-image, which has 180,000 text images and tagged pictures for each scene and is separated into training, validation, and test sets in the ratio 14:1:1, designated as text-train, text-valid, and text-test, respectively.

The MJSynth (MJ) synthetic scene dataset [58] is intended for scene text recognition applications. It comprises 9 million synthetic pictures that span 90,000 English words, with the first 1.5 million photos serving as the experiment's training set.

(2) Real scene dataset

The IIIT5K dataset [59] comprises 5000 actual scene word pictures. The training set consists of 2000 photos, whereas the test set consists of 3000 images. The experiment solely uses the test set.

The ICDAR 2015 (IC15) dataset [60] comprises 1500 real scene word pictures. There are 500 test photos and 1000 training images.

The ICDAR 2003 (IC03) dataset [61] consists of two datasets: character recognition (Char) and word recognition. The character recognition dataset has 6185 frames for the training set and 5430 frames for the test set, whereas the word recognition dataset contains 1157 frames for the training set and 1111 frames for the test set, with just two test sets utilized in the experiment.

The CUTE80 dataset [62] comprises 80 high-quality photos of real-world settings, as well as 288 cropped text areas used as test samples.

As demonstrated in Figure 9, the backdrop of real-world photographs is more complicated, with great contrast between light and dark, huge changes in the background environment, and partial occlusion and perspective effects. To produce a large number of labeled datasets for training the models in this study, we mix the real scene datasets with the Text Recognition Data Generator tool. These complicated scene photos can help the model improve its generalization abilities and tackle the text recognition challenge in difficult scenarios. Figure 9a,b depicts the synthetic and actual datasets, respectively.



Figure 9. Comparison between synthetic datasets and real datasets.

4.2. Experimental Setup

(1) Experimental environment: In the experimental verification, the computer hardware environment is Intel(R)Core(TM)i7-8700@3.20 GHz CPU, 16 GB RAM, NVIDIA GeForce GTX 2080 GPU, Nvidia, Santa Clara, CA, USA; the software environment is Windows 10 operating system; and the operation environment is Python 3.7, Pytorch 1.9, PaddleOcr 2.6, and CUDA Toolkit 10.2. Pytorch 1.9, PaddleOcr 2.6, and CUDA Toolkit 10.2 are used.

(2) Training details: By examining the influence of text picture size on the model, the input image's height and width are 150 and 500, respectively. BCN consists of four levels, each with eight attention heads. The balance factors λ_v , λ_l are set to one and one. The ADAM optimiser is used, with an initial learning rate of 1×10^{-4} that decreases to 1×10^{-5} after 6 epochs, a batch_size of 64, and an epoch_num size of 100.

(3) Evaluation metrics: The ENBC network's performance is evaluated at the pixel level using output pictures. To statistically measure the network's denoising performance, we use accuracy, recall, F-score, and intersection over union (IoU). Set the foreground text pixels to positive (pixel value 0) and the background pixels to negative (pixel value 255). Compare the pixels in the output image to the pixels at the same position in the label image. They can be classified as true positive (TP) if they match the foreground text pixels, false positive (FP) if they differ, true negative (TN) if they match the background text pixels, and false negative (FN) if they differ. Based on these four categories, the model may be evaluated using three metrics:

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

The F-score is the harmonic average of accuracy and recall:

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (16)$$

Recognition accuracy is a key performance statistic for assessing algorithms in scene recognition applications. Word accuracy (WA) is commonly employed as a word-level accuracy parameter in English scene text recognition to assess model performance. WA is the percentage of successfully identified characters to total test pictures. In this experiment, we run text recognition experiments on this model and the classical model on small and large sample datasets, respectively, and analyze the effects of character noise and background noise on recognition accuracy on different scene datasets, using recognition accuracy (Acc) as the model's performance evaluation metric.

$$Acc = \frac{Number_{correct}}{Number_{total}} \times 100\% \quad (17)$$

The ideal model best_accuracy on the evaluation set is chosen as the experimental model, and the experimental results in the synthetic scene dataset are as follows: The recognition improvement module properly recognizes 11,025 of the 11,250 photos in the test set, with a recognition accuracy (Acc) of 98% and an average time consumption of 0.038 s per image, meeting the real-time requirement of usage.

4.3. Eliminate Background and Character Self-Noise Interference

The ENBC network is trained on the text-train and text-valid datasets and tested on the text-test dataset. Table 1 lists the test results of the ENBC network, where P, R, and

F are abbreviations for the evaluation metrics precision, recall, and F-score, respectively. IoU stands for intersection over union. IoU evaluates the overlap between the predicted foreground region and the ground truth label region, with IoU values ranging from 0 to 1, where higher values are better. The calculation formula is:

$$IoU = \frac{TP}{TP + FP + FN} \quad (18)$$

Table 1. Test results of the ENBC network on text-test (%).

Dataset	P	R	F	IoU
Text-test	93.91	98.69	96.24	92.76

As shown in the table, the ENBC network achieved an F-score of 96.24% and an IoU value of 92.76% on the test set. This indicates that the ENBC network not only eliminates background noise in scene text images but also reduces the interference of character noise, thereby improving the accuracy and robustness of the recognition.

The test results of the ENBC network on the text-test, IIIT5K, IC15, IC03-char, IC03-word, and CUTE80 datasets for removing background noise and character noise are displayed in Figure 10 below to confirm the network's generalization performance on more datasets. The first row of each dataset displays the input images, which display problems including character distortion, uneven illumination, object occlusion when there is background noise, and comparable background and character textures.

	Background Noise	Character's Own Noise
Text-test	 ECO Ethan boning	 KEENER cafe lede
IIIT5K	 JULY RIVER Madness	 HOW THIS ACCOUNT
IC15	 HotPot KEEP BATTERY	 SKII CHARLIE strings
IC03-char	 G 7 P M R	 E S S S T
IC03-word	 NatWest Systems ID	 LIFE self Nigel
CUTE80	 WHITE ALBION academy	 BAKER Team ARTETA

Figure 10. Partial sample test visualization results.

4.4. Scene Text Recognition

Train the small sample dataset and the large sample dataset separately. The small sample dataset refers to the synthetic dataset generated using the Text Recognition Data Generator tool, while the large sample dataset is composed of both the synthetic dataset and the MJ dataset. The two types of datasets will be used separately to conduct text recognition experiments on both the proposed model and the classical model, using recognition accuracy as the performance evaluation metric for scene text recognition models. The impact of background noise and character noise on the accuracy of different scene datasets will be analyzed.

(1) Synthetic scene dataset experiment

The model presented in this research and the classical model are trained on the text-train dataset and tested on the text-test dataset, with results displayed in Table 2. When tested on a small sample dataset, this paper's model has an accuracy of 98%, and our suggested scene text recognition model, which eliminates the interference of backdrop and character noise, achieves a superior accuracy trade-off than other current models. Specifically, our suggested model outperforms attention-based integrated network for scene text recognition (ABINet), the scene text recognizer published in [19], by 7.42% on text-test. Overall, Acc obtains the maximum improvement of 16.28%, the lowest improvement of 4.2%, and an average increase of 10.41% when compared to other classical models. The experimental findings suggest that the model presented in this study may increase identification accuracy by reducing background and character noise in a synthetic scenario.

Table 2. Experimental results of synthetic scene dataset (%).

Methods	Training Dataset	Text-Test
CRNN	Text-train	82.91
RobustScanner	Text-train	88.28
SRN	Text-train	87.67
STAR-Net	Text-train	85.75
PREN	Text-train	86.04
ViTSTR-Small	Text-train	81.72
ABINet	Text-train	90.58
TrOCR	Text-train	93.80
MASTER	Text-train	91.57
Ours	Text-train	98.00

(2) Experiments on real scene datasets

Using text-train and the MJ dataset to train this model and the classical model, the model is tested on the actual scene datasets IIIT5K, IC15, IC03-Char, IC03-Word, and CUTE80, matching the findings shown in Table 3. The accuracy of this paper's model on the IIIT5K, IC15, IC03-Char, IC03-Word, and CUTE80 datasets is 96.10%, 80.47%, 87.22%, 83.75%, and 87.50%, respectively. It outperforms the ABINet scene text recognition technique by 9.65%, 5.78%, 4.89%, 0.02%, and 3.45%, respectively, thanks to the character feature augmentation effect provided by the ENBC network. Compared to other models, the recognition results of the proposed model in this paper show an improvement of at least 2.92%, 5.17%, and 2.56% on the IIIT5K, IC15, and IC03-Char datasets, respectively. However, the performance is slightly worse than the TrOCR model by 0.98% and 2.17% on the IC03-Word and CUTE80 datasets. Furthermore, our method outperforms CRNN [12], RobustScanner [63], SRN [64], STAR-Net [65], PREN [21], ViTSTR-Small [66], ABINet [19], TrOCR [67], and MASTER [68] with overall average accuracies of 72.22%, 79.03%, 80.75%, 78.79%, 81.62%, 79.62%, 82.25%, 83.53%, and 82.44% on public datasets, respectively, while

our model's total average rate can reach 87%. The experimental findings suggest that the model proposed in this study may increase recognition accuracy by removing background and character noise in the real picture.

Table 3. Experimental results of real scene dataset (%).

Methods	Training Dataset	Test Dataset					
		IIIT5K	IC15	IC03-Char	IC03-Word	CUTE80	Avg
CRNN	Text-train+MJ	79.38	61.35	72.22	76.31	71.87	72.22
RobustScanner	Text-train+MJ	83.16	72.1	77.96	81.46	80.49	79.03
SRN	Text-train+MJ	83.52	72.82	83.52	81.52	82.37	80.75
STAR-Net	Text-train+MJ	85.20	74.50	83.05	82.00	69.20	78.79
PREN	Text-train+MJ	87.05	73.45	81.38	80.61	85.61	81.62
ViTSTR-Small	Text-train+MJ	85.60	75.30	83.10	82.80	71.30	79.62
ABINet	Text-train+MJ	86.45	74.69	82.33	83.73	84.05	82.25
TrOCR	Text-train+MJ	84.35	74.22	84.66	84.73	89.67	83.53
MASTER	Text-train+MJ	93.18	70.52	83.76	82.67	82.07	82.44
Ours	Text-train+MJ	96.10	80.47	87.22	83.75	87.50	87.00

(3) Trade-off study

During the experiments, we conducted a comprehensive trade-off study on the average accuracy (Avg), parameters (Parameters), computational complexity (FLOPs), and inference speed (Speed). Table 4 shows the overall trade-offs under three comparison dimensions (accuracy vs. parameters, accuracy vs. FLOPs, accuracy vs. speed). Our model achieves an average accuracy of 87.00% with a speed of 38 milliseconds per image, having 39.7 M parameters and a computational complexity of 8.65×10^9 FLOPs. Specifically, compared to MASTER, our model has 4.55×10^9 fewer FLOPs, with 11.7 M more parameters and 31.55 milliseconds slower speed per image, but it outperforms the MASTER model by 4.56% in terms of average accuracy. As for the parameters of our model, it can be seen from the table that compared to the STAR-Net model, it reduces 9.2 M parameters, and compared to the SRN model, it reduces 17.6 M parameters. Additionally, the parameters, computational complexity, and inference speed of our model are significantly better than those of TrOCR, and its average accuracy also outperforms other models. Therefore, the proposed ENBC model is at or near the optimal point in the trade-offs between accuracy vs. speed, accuracy vs. parameters, and accuracy vs. FLOPs, with shorter runtime and lower computational overhead.

Table 4. Model average accuracy (Avg.), parameters, computational requirements (FLOPs), and speed.

Methods	Avg	Parameters (1×10^6)	Flops (1×10^9)	Speed (ms/Image)
CRNN	72.22	8.5	1.4	3.7
RobustScanner	79.03	20.6	5.0	16.8
SRN	80.75	57.3	10.8	18.8
STAR-Net	78.79	48.9	10.7	8.8
PREN	81.62	20.0	3.45	29.5
ViTSTR-Small	79.62	21.5	4.6	9.5
ABINet	82.25	36.7	8.3	33.9
TrOCR	83.53	558.0	15.2	318.0
MASTER	82.44	28.0	13.2	6.45
Ours	87.00	39.7	8.65	38.0

(4) Model failure analysis

Figures 11 and 12 demonstrate several accurate and incorrect recognition instances from this paper's model on the scene dataset. Among them, the first row of the table is the input image, the second row is the reconstructed image through the ENBC network to

enhance the character features, the third row is the real result of the recognition expectation, and the fourth row is the recognition result through the model proposed in this paper. Furthermore, the black type in the table's recognition results represents successfully identified characters, the red font indicates erroneously recognized characters, and the “_” indicates loss of recognized characters. The model in this research performs admirably in demanding text pictures, as seen by the accurate recognition examples in Figure 11. Specifically, the model in this paper excels at recognizing scene text images with character textures similar to the background texture, complex backgrounds, and strong noise interference from the characters themselves, as well as some scene text images that are difficult to recognize even by humans. For example, recognizing the proper example of “American” is tough since the font color and background color are quite close to the human eye, but this paper model can detect the true impact. When recognizing “HOW” in the correct example, there is a lot of noise in the character itself, but using the ENBC network in this study, it can accomplish the impact of denoising and improve recognition accuracy.

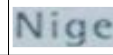
	Successful Recognition Examples							
Input Image								
Reconstructed Image	GILLIAM	Crestline	cafe	18	Nagashima	American	HOW	Nigel
Real Result	GILLIAM	Crestline	cafe	18	Nagashima	American	HOW	Nigel
Recognition Result	GILLIAM	Crestline	cafe	18	Nagashima	American	HOW	Nigel

Figure 11. Correct example process of scene image recognition.

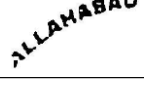
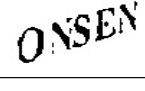
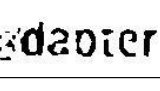
	Wrong Recognition Examples				
Input Image					
Reconstructed Image					
Real Result	ALLAHABAD	O	ONSEN	NOKIA	adapter
Recognition Result	_LLAHABA O	n	O VSEN	A OKIA	_daoier

Figure 12. Example process of scene image recognition error.

Figure 12 also contains instances of recognition mistakes, which may be broken down into three categories. First, the curvature of the text is too great, as seen by the semi-ellipse shape in the “ALLAHABAD” picture, causing certain pixels to be true throughout the model identification process, impacting the right recognition results. Second, there are too many missing characters, such as the “o” in the image, which was incorrectly identified as an “n” due to a significant lack of the lower portion of the character. Third, major blurring and big exposure, such as “adapter” were misread as “_daoier”, owing to the exposure rate being too strong, immediately lost the initial letter, and serious blurring caused by the “pt” was misclassified as “oi”. These mistake instances also indicate the suggested model’s future research directions. Finally, Figure 13 depicts some sample instances of the proposed model on the scene dataset, categorized based on the recognition effect, with

Figure 13a displaying properly identified samples and Figure 13b displaying erroneously recognized samples.



Figure 13. Some sample examples.

To summarize, the model presented in this research can successfully reduce background and character noise in natural scene photos while also improving identification performance when compared to the traditional model.

(5) Adaptation Strategy

Considering that the experiments in this paper require discussing adaptation strategies to provide a comprehensive evaluation, we adopted three domain-adaptation fine-tuning approaches in our ENBC model. First, the decoder (multi-level character feature fusion module) is responsible for merging multi-scale features extracted by the encoder with low-level details to eliminate background noise and generate the reconstructed image. Secondly, the recognition enhancement module improves the accuracy of character sequence prediction by modeling the context using a bidirectional transformer. Lastly, joint fine-tuning (decoder + recognition module) adjusts the parameters of both the decoder and the recognition module simultaneously, maximizing the domain adaptation effect. In the text-train + MJ pre-trained ENBC model, we used IC15 as the primary evaluation metric, and IIIT5K and CUTE80 as auxiliary metrics, to validate the model's generalization ability. Table 5 presents the design of the fine-tuning strategies (the explanation for the table header can be found in the table notes), and Table 6 shows the comparison of performance improvements.

Table 5. Design of fine-tuning strategies.

Strategy	Frozen Modules	Trainable Modules	Learning Rate	Training Data
Strategy A	Encoder	Decoder	1×10^{-5}	IC15 (1000)
Strategy B	Encoder + Decoder	Recognition Module	1×10^{-5}	IC15 (1000)
Strategy C	Encoder	Decoder + Recognition Module	1×10^{-5}	IC15 (1000)

According to the data presented in Table 6, strategy B (fine-tuning only the recognition module) improved the accuracy of IC15 by 5.15%, validating the sensitivity of the recognition module to domain adaptation. Strategy C (joint fine-tuning) further improved the accuracy to 86.10%, with minimal impact on other datasets, indicating that global optimization is more effective. It is worth noting that the IC15 training set contains only 1000 images, so semi-supervised learning (using unlabeled real data) can effectively alleviate the issue of insufficient data and enhance the model's performance.

Table 6. Comparison of performance improvements.

Strategy	IC15 Acc (%)	IIIT5K Acc (%)	CUTE80 Acc (%)
Baseline	80.47	96.10	87.50
Strategy A	83.20 (+2.73)	95.85 (−0.25)	87.30 (−0.20)
Strategy B	85.62 (+5.15)	96.05 (−0.05)	87.45 (−0.05)
Strategy C	86.10 (+5.63)	96.15 (+0.05)	87.60 (+0.10)

5. Conclusions

This research presents an approach for scene text recognition based on ENBC. To begin, the dataset required by ENBC is built through preprocessing, and a synthetic scene dataset, text-image, including 180,000 scene text images and label pictures, is created. The ENBC network was then proposed, in which the encoder obtained features of different scales via ASPP dilated convolution with different expansion rates to expand the receptive field, allowing it to more effectively capture the character feature area, eliminate character noise interference, and improve character shape features. The decoder combined the high-level character feature information after upsampling with the low-level character feature information in the backbone network, separated the foreground characters from background interference, removed background noise, and produced the reconstructed picture. Finally, the recognition improvement module improves character context modeling by taking into account both forward and backward information from the text sequence. Experimental findings on synthetic and actual datasets demonstrate that, when compared to other popular approaches, ENBC has the highest accuracy on scene datasets and an outstanding recognition effect for text recognition. This technique effectively addresses the issue that previous models fail to properly reduce background noise and character noise when directly processing scene photos. However, in practice, the model in this study does not fully utilize the semantic information of characters due to the issue that the curvature of the text is too great and the characters are absent from the natural scene image. When visual information is absent and certain characters are erroneously identified, we will continue to investigate context semantic information modeling of the visual model's output in order to increase recognition accuracy.

Author Contributions: Conceptualization, S.T., S.L., Z.J. and K.L.; Methodology, Y.C.; Software, Y.C.; Validation, Y.C.; Formal analysis, Y.C.; Investigation, Y.C.; Resources, Y.C.; Data curation, Y.C.; Writing—original draft, Y.C.; Visualization, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Xi'an Science and Technology Plan Project with the grant number [23KGDW0032-2022].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liang, Y.; Li, X. Reassembling shredded document stripes using wordpath metric and greedy composition optimal matching solver. *IEEE Trans. Multimedia* **2020**, *22*, 1168–1181. [[CrossRef](#)]
2. Zhang, J.; Sang, J.; Xu, K.; Wu, S.; Zhao, X.; Sun, Y.; Hu, Y.; Yu, J. Robust CAPTCHAs towards malicious OCR. *IEEE Trans. Multimedia* **2021**, *23*, 2575–2587. [[CrossRef](#)]

3. Tian, S.; Lu, S.; Su, B.; Tan, C.L. Scene text recognition using co-occurrence of histogram of oriented gradients. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 912–916.
4. Su, B.; Lu, S.; Tian, S.; Lim, J.H.; Tan, C.L. Character recognition in natural scenes using convolutional co-occurrence hog. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 2926–2931.
5. Su, B.; Lu, S. Accurate scene text recognition based on recurrent neural network. In Proceedings of the Asian Conference on Computer Vision, Singapore, 1–5 November 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 35–48.
6. Casey, R.; Nagy, G. Recognition of printed Chinese characters. *IEEE Trans. Electron. Comput.* **1966**, EC-15, 91–101. [\[CrossRef\]](#)
7. Zhou, S.S.; Chen, Q.C.; Wang, X.L.; Guo, X.; Li, H. An empirical evaluation on HIT-OR3C database. In Proceedings of the International Conference on Document Analysis and Recognition, Beijing, China, 18–21 September 2011; IEEE Computer Society Press: Los Alamitos, CA, USA, 2011; pp. 1150–1154.
8. Qu, X.W.; Xu, N.; Wang, W.Q.; Lu, K. Similar handwritten Chinese character recognition based on adaptive discriminative locality alignment. In Proceedings of the 14th IAPR International Conference on Machine Vision Applications, Tokyo, Japan, 18–22 May 2015; IEEE Computer Society Press: Los Alamitos, CA, USA, 2015; pp. 130–133.
9. Wu, X.; Chen, Q.; Xiao, Y.; Li, W.; Liu, X.; Hu, B. LCSegNet: An efficient semantic segmentation network for large-scale complex Chinese character recognition. *IEEE Trans. Multimed.* **2020**, 23, 3427–3440. [\[CrossRef\]](#)
10. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, 86, 2278–2324. [\[CrossRef\]](#)
11. Sheng, F.; Zhai, C.; Chen, Z.; Xu, B. End-to-end chinese image text recognition with attention model. In Proceedings of the International Conference on Neural Information Processing, Guangzhou, China, 14–18 November 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 180–189.
12. Shi, B.; Bai, X.; Yao, C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, 39, 2298–2304. [\[CrossRef\]](#)
13. He, T.; Huang, W.; Qiao, Y.; Yao, J. Text-attentional convolutional neural network for scene text detection. *IEEE Trans. Image Process.* **2016**, 25, 2529–2541. [\[CrossRef\]](#)
14. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, Pennsylvania, PA, USA, 25–29 June 2006; ACM: New York, NY, USA, 2006; pp. 369–376.
15. Liao, M.; Zhang, J.; Wan, Z.; Xie, F.; Liang, J.; Lyu, P.; Yao, C.; Bai, X. Scene text recognition from two-dimensional perspective. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8714–8721.
16. Cheng, Z.; Xu, Y.; Bai, F.; Niu, Y.; Pu, S.; Zhou, S. Aon: Towards arbitrarily-oriented text recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5571–5579.
17. Wang, T.; Zhu, Y.; Jin, L.; Luo, C.; Chen, X.; Wu, Y.; Wang, Q.; Cai, M. Decoupled attention network for text recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12216–12224.
18. Nguyen, N.; Tran, V.; Tran, M.-T.; Ngo, T.D.; Nguyen, T.H.; Hoai, M. Dictionary-guided scene text recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7383–7392.
19. Fang, S.; Xie, H.; Wang, Y.; Mao, Z.; Zhang, Y. Read like humans: Autonomous, bidirectional and iterative language modelling for scene text recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 7098–7107.
20. Tounsi, M.; Moalla, I.; Pal, U.; Alimi, A.M. Arabic and Latin scene text recognition by combining handcrafted and deep-learned features. *Arab. J. Sci. Eng.* **2022**, 47, 9727–9740. [\[CrossRef\]](#)
21. Yan, R.; Peng, L.; Xiao, S.; Yao, G. Primitive representation learning for scene text recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 284–293.
22. Zhang, Z.; Xu, Y.; Liu, C.-L. Natural scene character recognition using robust pca and sparse representation. In Proceedings of the 2016 12th IAPR Workshop on Document Analysis Systems (DAS), Santorini, Greece, 11–14 April 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 340–345.
23. Tian, S.; Bhattacharya, U.; Lu, S.; Su, B.; Wang, Q.; Wei, X.; Lu, Y.; Tan, C.L. Multilingual scene character recognition with co-occurrence of histogram of oriented gradients. *Pattern Recognit.* **2016**, 51, 125–134. [\[CrossRef\]](#)
24. Su, B.; Lu, S. Accurate recognition of words in scenes without character segmentation using recurrent neural network. *Pattern Recognit.* **2017**, 63, 397–405. [\[CrossRef\]](#)

25. Liang, M.; Hu, X. Recurrent convolutional neural network for object recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; IEEE: New York, NY, USA, 2015; pp. 3367–3375.
26. Wang, J.; Hu, X. Gated recurrent convolution neural network for ocr. In *Advances in Neural Information Processing Systems*; Curran Associates: New York, NY, USA, 2017; Volume 30.
27. Albahli, S.; Nawaz, M.; Javed, A.; Irtaza, A. An improved faster-RCNN model for handwritten character recognition. *Arab. J. Sci. Eng.* **2021**, *46*, 8509–8523.
28. Zhang, Z.; Wang, H.; Liu, S.; Xiao, B. Consecutive convolutional activations for scene character recognition. *IEEE Access* **2018**, *6*, 35734–35742.
29. Zhang, Y.; Nie, S.; Liu, W.; Xu, X.; Zhang, D.; Shen, H.T. Sequence-to-sequence domain adaptation network for robust text image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; IEEE: New York, NY, USA, 2019; pp. 2740–2749.
30. Liu, Z.; Li, Y.; Ren, F.; Goh, W.L.; Yu, H. Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; AAAI: Menlo Park, CA, USA, 2018.
31. Bhunia, A.K.; Banerjee, P.; Konwer, A.; Bhowmick, A.; Roy, P.P.; Pal, U. Word level font-to-font image translation using convolutional recurrent generative adversarial networks. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 3645–3650.
32. Bhunia, A.K.; Das, A.; Bhunia, A.K.; Kishore, P.S.R.; Roy, P.P. Handwriting recognition in low-resource scripts using adversarial learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4767–4776.
33. Wu, S.; Zhai, W.; Cao, Y. Pixtextgan: Structure aware text image synthesis for license plate recognition. *IET Image Process.* **2019**, *13*, 2744–2752. [[CrossRef](#)]
34. Wang, Y.; Lian, Z.; Tang, Y.; Xiao, J. Boosting scene character recognition by learning canonical forms of glyphs. *Int. J. Doc. Anal. Recognit. (IJ DAR)* **2019**, *22*, 209–219.
35. Lin, Q.; Liang, L.; Huang, Y.; Jin, L. Learning to generate realistic scene chinese character images by multitask coupled gan. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Guangzhou, China, 23–26 November 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 41–51.
36. Bonechi, S.; Bianchini, M.; Scarselli, F.; Andreini, P. Weak supervision for generating pixel-level annotations in scene text segmentation. *Pattern Recognit. Lett.* **2020**, *138*, 1–7.
37. Wang, C.; Zhao, S.; Zhu, L.; Luo, K.; Guo, Y.; Wang, J.; Liu, S. Semi-supervised pixel-level scene text segmentation by mutually guided network. *IEEE Trans. Image Process.* **2021**, *30*, 8212–8221. [[CrossRef](#)]
38. Chaitra, Y.L.; Dinesh, R.; Gopalakrishna, M.T.; Prakash, B.V.A. Deep-CNNLT: Text localization from natural scene images using deep convolution neural network with transfer learning. *Arab. J. Sci. Eng.* **2022**, *47*, 9629–9640.
39. Xu, X.; Zhang, Z.; Wang, Z.; Price, B.; Wang, Z.; Shi, H. Rethinking text segmentation: A novel dataset and a text specific refinement approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2021; IEEE: New York, NY, USA, 2021; pp. 12045–12055.
40. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoderdecoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 801–818.
41. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; ACM: New York, NY, USA, 2015; pp. 448–456.
42. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; IEEE: New York, NY, USA, 2017; pp. 1251–1258.
43. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Xiao, B. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364.
44. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE international Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1520–1528.
45. Liu, C.; Chen, L.-C.; Schroff, F.; Adam, H.; Hua, W.; Yuille, A.L.; Li, F.-F. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 82–92.
46. Yan, J.; Cheng, Y.; Wang, Q.; Liu, L.; Zhang, W.; Jin, B. Transformer and graph convolution-based unsupervised detection of machine anomalous sound under domain shifts. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *8*, 2827–2842.
47. Yan, J.; Wang, X.; Cai, J.; Qin, Q.; Yang, H.; Wang, Q.; Cheng, Y.; Gan, T.; Jiang, H.; Deng, J.; et al. Medical image segmentation model based on triple gate MultiLayer perceptron. *Sci. Rep.* **2022**, *12*, 6103.

48. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
49. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [PubMed]
50. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
51. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
52. Lyu, P.; Liao, M.; Yao, C.; Wu, W.; Bai, X. 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), Munich, Germany, 8–14 September 2018; pp. 67–83.
53. Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vis.* **2016**, *116*, 1–20.
54. Fisher, Y.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *Int. Conf. Learn. Represent.* **2016**, *18*, 554–568.
55. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
56. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNet v2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 4510–4520.
57. Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; Bai, X. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2035–2048.
58. Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. [2022–07–25]. Available online: <https://arxiv.org/abs/1406.2227> (accessed on 10 December 2024).
59. Mishra, A.; Alahari, K.; Jawahar, C.V. Scene Text Recognition Using Higher Order Language Priors. [2022–07–25]. Available online: <https://inria.hal.science/hal-00818183/document> (accessed on 10 December 2024).
60. Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V.R.; Lu, S.; et al. ICDAR 2015 competition on robust reading. In Proceedings of the 13th International Conference on Document Analysis and Recognition, Tunis, Tunisia, 23–26 August 2015; IEEE Computer Society Press: Los Alamitos, CA, USA, 2015; pp. 1156–1160.
61. Lucas, S.M.; Panaretos, A.; Sosa, L.; Tang, A.; Wong, S.; Young, R.; Ashid, K.; Nagai, H.; Okamoto, M.; Yamamoto, H.; et al. ICDAR 2003 robust reading competitions: Entries, results, and future directions. *Int. J. Doc. Anal. Recognit. (IJ DAR)* **2005**, *7*, 105–122.
62. Risnumawan, A.; Shivakumara, P.; Chan, C.S.; Tan, C.L. A robust arbitrary text detection system for natural scene images. *Expert Syst. Appl.* **2014**, *41*, 8027–8048.
63. Yue, X.; Kuang, Z.; Lin, C.; Sun, H.; Zhang, W. RobustScanner: Dynamically enhancing positional clues for robust text recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 135–151.
64. Yu, D.; Li, X.; Zhang, C.; Liu, T.; Han, J.; Liu, J.; Ding, E. Towards accurate scene text recognition with semantic reasoning networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 12113–12122.
65. Liu, W.; Chen, C.; Wong, K.-Y.; Su, Z.; Han, J. STAR-Net: A SpaTial attention residue network for scene text recognition. In Proceedings of the British Machine Vision Conference, York, UK, 19–22 September 2016; Volume 2, p. 7.
66. Atienza, R. Vision transformer for fast and efficient scene text recognition. In *Document Analysis and Recognition—ICDAR 2021*; Springer: Cham, Switzerland, 2021; pp. 319–334.
67. Li, M.; Lv, T.; Chen, J.; Cui, L.; Lu, Y.; Florencio, D.; Zhang, C.; Li, Z.; Wei, F. Trocr: Transformer-based optical character recognition with pre-trained models. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 13094–13102.
68. Lu, N.; Yu, W.; Qi, X.; Chen, Y.; Gong, P.; Xiao, R.; Bai, X. Master: Multi-aspect non-local network for scene text recognition. *Pattern Recognit.* **2021**, *117*, 107980.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.