information

MDPI

Article

Scene Text Recognition Based on Improved CRNN

Wenhua Yu [1,2], Mayire Ibrayim [1,2,*] and Askar Hamdulla [1,3]

[1] College of Information Science and Engineering, Xinjiang University, Urumqi 830017, China; yuwenhua@stu.xju.edu.cn (W.Y.); askar@xju.edu.cn (A.H.)
[2] Xinjiang Key Laboratory of Signal Detection and Processing, Urumqi 830017, China
[3] Xinjiang Key Laboratory of Multilingual Information Technology, Urumqi 830017, China
* Correspondence: mayire401@xju.edu.cn; Tel.: +86-133-1988-9043

Abstract: Text recognition is an important research topic in computer vision. Scene text, which refers to the text in real scenes, sometimes needs to meet the requirement of attracting attention, and there is the situation such as deformation. At the same time, the image acquisition process is affected by factors such as occlusion, noise, and obstruction, making scene text recognition tasks more challenging. In this paper, we improve the CRNN model for text recognition, which has relatively low accuracy, poor performance in recognizing irregular text, and only considers obtaining text sequence information from a single aspect, resulting in incomplete information acquisition. Firstly, to address the problems of low text recognition accuracy and poor recognition of irregular text, we add label smoothing to ensure the model's generalization ability. Then, we introduce the smoothing loss function from speech recognition into the field of text recognition, and add a language model to increase information acquisition channels, ultimately achieving the goal of improving text recognition accuracy. This method was experimentally verified on six public datasets and compared with other advanced methods. The experimental results show that this method performs well in most benchmark tests, and the improved model outperforms the original model in recognition performance.

Keywords: CRNN; text recognition; label smoothing; language model; deep learning

check for updates

Citation: Yu, W.; Ibrayim, M.; Hamdulla, A. Scene Text Recognition Based on Improved CRNN. *Information* **2023**, *14*, 369. https://doi.org/10.3390/info14070369

Academic Editor: Shmuel Tomi Klein

Received: 27 April 2023
Revised: 17 June 2023
Accepted: 27 June 2023
Published: 28 June 2023

1. Introduction

Text recognition is an important direction in the field of computer vision. With the continuous development of deep learning fields such as computer vision, pattern recognition, and machine learning, scene text recognition of deep learning has been developed on this basis. Text recognition can be divided into two branches according to recognition algorithms: segmentation-based recognition algorithms and recognition algorithms that do not require segmentation. The segmentation-based natural scene text recognition algorithm usually needs to locate the location of each character contained in the input text image, identify each character through a single character recognizer, and then combine all the characters into a string sequence to obtain the final recognition result. Natural scene text recognition algorithms without segmentation aim to treat the entire text line as a whole and directly map the input text image to a sequence of target strings, thus avoiding the disadvantages and performance limitations of single character segmentation, which is also the current mainstream approach [1]. In the process of text recognition, a series of labels are usually predicted, and the whole recognition process can be regarded as a sequence recognition problem [2]. The CRNN [2] algorithm in sequence recognition without segmentation is a neural network that integrates feature extraction, sequence modeling, and transcription. The feature map is first extracted using a convolutional neural network (CNN), and then the feature dependencies are captured using a recurrent neural network (RNN), the features are predicted, and the output prediction distribution is fed to connectionist temporal classification (CTC) [3] for processing, and then the final text sequence is output. Because RNN models are an important branch of deep neural networks, they are designed primarily to process sequences. To learn text sequences directly using RNN models, there are dependencies: the mapping relationship between input and output sequences needs to be labeled in advance, and the mapping relationship between input and output sequences is a one-to-one correspondence. Because text and speech signals are continuous signals, there are segmentation difficulties, and the volume of text recognition data is in the millions; it is also costly, time-consuming, and impractical to achieve segmentation and annotation. Therefore, it is not applicable to apply an RNN directly to text recognition, so the CRNN network adds the CTC proposed by Alex Graves et al. after the RNN [3] to make RNN applicable to text recognition. The CTC algorithm is an end-to-end training method for RNNs. It extends the output layer of an RNN by: converting the data dependency of "segmenting" and label mapping relationships to extracting features according to a sliding time window, transforming the input–output relationship from one-to-one to many-to-one; adding blank characters, and performing deduplication and blank character removal operations on consecutive identical characters in the sequence output; reducing complexity and increasing speed by drawing on the forward and backward algorithms of the hidden Markov model (HMM) to compute the loss function; and using dynamic programming to compute the training paths, avoiding impractical exhaustive methods or violent enumeration. The decoding process of CTC maps the path generated by CTC into a final sequence. Combining these features, an example of the final sequence mapped after deduplication and removal of a whitespace character using CTC is shown in Figure 1.

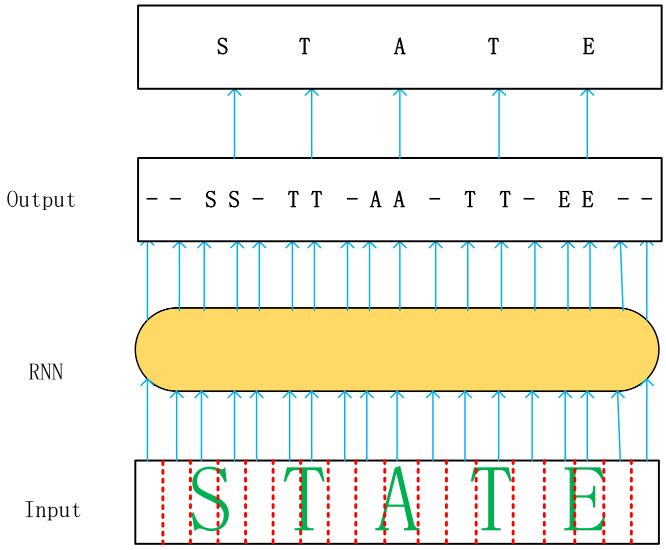

Figure 1. Conversion correspondence diagram.

CRNN models have low text recognition accuracy [4,5], poor recognition of irregular text, and incomplete information acquisition by acquiring text sequence information from a single level. The commonly used improvement schemes are mostly focused on two networks, CNN and RNN, to analyze the inadequate feature extraction of the network, the presence of gradient explosion disappearance of the network [6], and the poor recognition of indefinitely long text sequences. Targeted replacements have been made to improve feature extraction networks, replacing recurrent neural networks with long short-term memory networks [7] or adding residual modules. Although better performance is achieved, there is the problem that the acquisition is not comprehensive and is limited to the text domain only. Therefore, based on this, this paper takes CTC and the whole CRNN as the entry point, adds a label smoothing strategy, introduces the smoothing loss function in the field of speech recognition into the field of text recognition, and adds a language model, taking into account the acquisition of information from various aspects to achieve the improvement of recognition accuracy.

The main contributions of this paper are as follows. Firstly, for the low accuracy of recognition results, data labeled as hard labels will introduce noise and loss of information, leading to poor generalization of the model and recognition results being easily affected. Secondly, adding label smoothing to obtain soft labels, which carry more information, are more robust to noise, and improve the generalization ability of the model. Thirdly, after combining CTC with label smoothing, the loss function after label smoothing is redefined. Finally, the language model is connected after the CRNN model, and the CRNN prediction results are input to the language model as the prior knowledge of the language model, so that the complementary nature of visual information and language information can be used to obtain text information from multiple levels to improve the accuracy of text recognition further and achieve relatively high recognition accuracy on the six test sets. As the improved model is divided into two main parts, namely the language model and the CRNN with label smoothing, the latter can be replaced with other visual models. Therefore, the two parts are relatively independent.

The rest of the paper is organized as follows. Section 2 summarizes the relevant research in this field, with a focus on text recognition in the field of deep learning. In Section 3, the CRNN recognition model, which combines label smoothing and the language model, is introduced in detail. Experimental results and corresponding discussions are provided in Section 4. The paper concludes with a summary in Section 5.

2. Related Work

As the field of deep learning continues to develop, the direction of scene text recognition has also been developed, and many researchers have proposed many new and relevant recognition algorithms. The CRNN [2] network model, proposed in 2015, is a classical model in the field of text recognition, combining CNN, RNN, and CTC [3] to perform text recognition from the perspective of text sequences and avoid the limitation of accurate slicing. First, the input image is converted into a grayscale map, feature extraction is performed using CNN, contextual information is learned using RNN, and finally the network is optimized using CTC to solve the text alignment problem. In 2016, the RARE [8] algorithm was proposed, combining spatial transformation networks and sequence recognition networks for curved text correction recognition. In 2016, the R2AM [9] algorithm was proposed, which for the first time introduced an attention mechanism into the field of text recognition and implemented soft feature selection in the decoding process to utilize image features better. The STAR-Net [10] network, proposed in 2016, uses spatial transformation to remove text distortions and uses residual convolution blocks to construct feature extractors, particularly effective in distortion-rich scenes of text. The GRCNN [11] model was proposed in 2017. It introduces a gating strategy in the recurrent convolution layer (RCL) to control the context information and balance the transmission of forward and recursive information. By combining GRCNN with bidirectional LSTM, the entire network can be trained end-to-end, thereby effectively recognizing text information in images In 2018, the paper [12] proposed an optical character recognition system called Rosetta. The system consists of two stages: Text detection stage, based on the Faster-RCNN model, detects text regions in the image; Character recognition model based on fully convolutional networks processes the detected text regions and recognizes the text content. Benchmark [13], presented in 2019, provides a disassembled analysis of the model for the STR task, which helps researchers gain insight into the model and make improvements to existing models. The semantically enhanced codec architecture for recognizing low-quality scene text was proposed in SEED [14] in 2020. As transformers continue to evolve and transformers as decoders become more common in STR tasks, recognition tasks are beginning to focus on more than just recognition accuracy. ViTSTR [15], proposed in 2021, uses a simple single-stage model architecture built on a computationally and parametrically efficient visual transformer (ViT) to maximize accuracy, speed, and computational efficiency. TRBA [16] makes full use of real data through data augmentation, collecting unlabeled data and introducing semi-supervised and self-supervised improvements to the model, moving in the direction of text recognition for scenes with fewer labels. Ref. [17] proposed cascaded attention networks using three attention modules, from horizontal continuity properties, contextual information, and two-bit visual distribution, addressing the drift phenomenon in encoding and decoding architectures. Text is Text [18] uses a single model to deal with scene text recognition (STR) and handwritten text recognition (HTR) for handwritten text, introducing a knowledge distillation (KD)-based framework to deal with the combination of STR and HTR, while proposing four distillation losses specifically designed to deal with the unique features of the aforementioned text recognition. Proposed in 2022, character-context decoupling [19] focuses on open-set text recognition tasks and proposes a character-context decoupling framework to alleviate the problem of confounding effects of contextual information over visual information of individual characters by separating contextual and character-visual information, with good results on both open and closed datasets. With the continuous development of text recognition algorithms, the addition of attention mechanisms and various encoding and decoding networks has achieved good results in terms of recognition accuracy. However, the overall network models have become more complex and less easy to understand. Typically, these models require high-performance experimental equipment. Therefore, compared to other models, this paper proposes a text recognition approach based on the classic CRNN network model, which has a clear and understandable structure. It achieves good recognition results while requiring lower experimental equipment requirements.

3. Methods

From the overall network structure diagram in Figure 2, it can be seen that the CRNN network with label smoothing (LS) added is used as the visual model, and the images with text areas cropped out are fed into the visual model, and features are extracted from the input images by the CNN in the visual model to obtain the feature maps. Next, the feature sequence is fed into a two-layer BiLSTM network for prediction (BiLSTM is an improvement over the bidirectional RNN network). The BiLSTM network learns the feature vectors in the sequence and outputs a predicted label distribution. Using a modified CTC loss, the series of label distributions obtained in the RNN are converted into a final label order as the prediction result of the visual model, and the prediction result is fed into the language model bidirectional cloze network (BCN). After a multi-headed attention mechanism and a feed-forward network, it is then subjected to linear variation to obtain the language model prediction results; the final output is the text STATE in the picture. Figure 3 is a flow chart of the improved CRNN network recognition process. The serial numbers ① and ② are the improved parts of the CRNN network, and the improvement points and the overall recognition process can be clarified by the color change. The main network structure used is a three-layer structure consisting of convolutional layers, recurrent layers, and transcription layers, using CNN+RNN+CTC. The convolutional layers consist of 7 layers of convolutional neural networks, and the basic structure uses the VGG structure. First, the input image is converted to a grayscale image, and then the grayscale image is resized to a size of W*32, with a fixed height. In the third and fourth pooling layers, a kernel size of 1*2 (rather than 2*2) is used to pursue the true aspect ratio, and a batchnorm (BN) layer is introduced to speed up convergence. The feature sequence obtained from the convolutional layers is predicted by the recurrent layers using an RNN (the RNN network can be a type of recurrent neural network such as LSTM or GRU) to predict the label distribution of the feature sequence, which represents the probability distribution of the true label of each time step in the feature sequence. The feature maps extracted by the CNN are split by column, and each column of 512-dimensional features is input into two layers of 256-unit bidirectional LSTMs for classification. The label distribution obtained from the recurrent layers is converted into the final recognition result by the transcription layer using CTC. The CTC algorithm performs deduplication and other operations to obtain the final recognition result, and label smoothing is added to this process. The recognition result is used as prior knowledge and sent to the language model for character correction, and the final result is output.

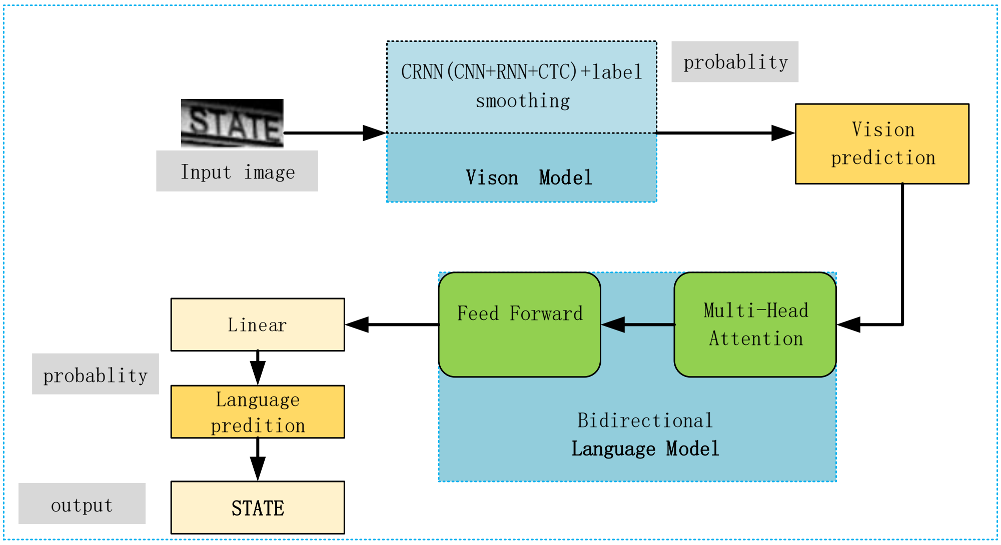

Figure 2. Improved overall network structure.

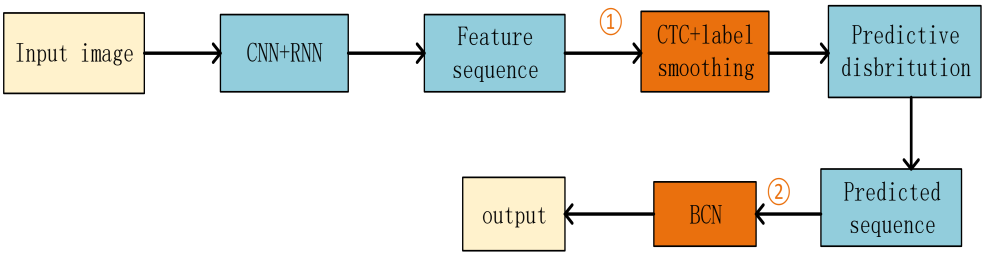

Figure 3. Overall network flow diagram.

As can be seen in Figure 1, in the text recognition task, the role of the CTC transcription layer is to take the output predicted text sequence from the RNN as input and transform it into a label sequence. Mathematically, transcription is used to find the tag sequence with the biggest probability based on the prediction [2]. The probabilities of the label sequences use the conditional probabilities from CTC [3].

In the CRNN network model, from the input image to the output text recognition result, it can be considered that the model applies feature information from the visual aspect to the text sequence recognition without applying feature information from other modalities. Considering that the overall recognition accuracy of the CRNN network model is still low and that relying on visual information from a single modality for text recognition is not informative enough, it is necessary to add other auxiliary information to improve the overall recognition accuracy based on the visual model effect. At this stage, linguistic features have been used to some effect in the text domain. Linguistic features refer to the consideration of the context between characters to infer the class of that character, rather than based on the glyphic features of the character. In this paper, we choose to add a linguistic model, a network model to obtain information relying on both visual and linguistic features rather than on visual features alone, and more comprehensive information acquisition, which helps to improve the accuracy of text recognition. The data used for text recognition are labeled data. Taking the English alphabet as an example: if it is not case sensitive, the English alphabet has 26 letters, and then considering the Arabic numerals 0–9, the whole data eventually correspond to 37 characters, so the text recognition problem is essentially a multivariate classification problem, with the option of adding label smoothing. Label smoothing is a regularization technique that ensures the generalization ability of the model and improves the resistance of the model to interference.

3.1. Label Smoothing (LS)

Label smoothing is a method of model regularization that can significantly improve the generalization ability and learning speed of multi-class neural networks, and is typically used to prevent model overfitting [20]. Smoothing labels in this way prevents networks from becoming overconfident and has been used in many state-of-the-art models, including image classification, language translation, and speech recognition. In addition to improving generalization, label smoothing improves model calibration and can significantly improve beam search [21].

Considering applying label smoothing to text recognition, since the CTC component of the CRNN model comes from the direction of speech recognition, label smoothing is a general method of improving generalization ability by adding label noise, which has the effect of penalizing low-entropy output distributions (i.e., overconfident predictions). In the classification process, one-hot encoding, which is commonly used, has poor generalization ability and tends to believe too much in labels, assuming that the differences between each category are large, which is actually difficult to achieve [22]. To address the issues with one-hot encoding, label smoothing is proposed [21], and the calculation formula is shown in (5).

$$\text{Label Smoothing} = \text{onehot} * (1 - \mathcal{E}) + \frac{\mathcal{E}}{c}, \tag{1}$$

In Equation (1), one-hot is the unique hot encoding variable of the tag, such as [0, 1], [1, 0, 0], etc., where ε is a hyper-parameter less than 1 and greater than 0 and c is the number of categories. The default value of the parameter ε is 0.1. When one-hot is [0, 1], the code becomes [0.05, 0.95] after label smoothing according to the formula. The process of adding label smoothing is achieved by adding a regularization term to the CTC objective function, which consists of KL scattering between the predictive distribution, P, of the network and the uniform distribution, U on the labels [23].

$$\mathrm{L}(\theta_{online}) \triangleq (1-\alpha)L_{CTC} + \alpha \textstyle\sum_{t=1}^{T} D_{KL}(P_t|U), \tag{2}$$

The adjustable parameter, α, in Equation (2) is used to balance the weight regularization term and CTC loss. From Equation (2), it can be intuitively seen that the whole loss function after adding label smoothing contains the CTC part and KL scatter part. Both parts are related to α, and when α takes 0 then it becomes L_{CTC}, and when α takes 1 then it becomes $D_{KL}(P_t|U)$. The CRNN model diagram after adding label smoothing is shown in Figure 4.

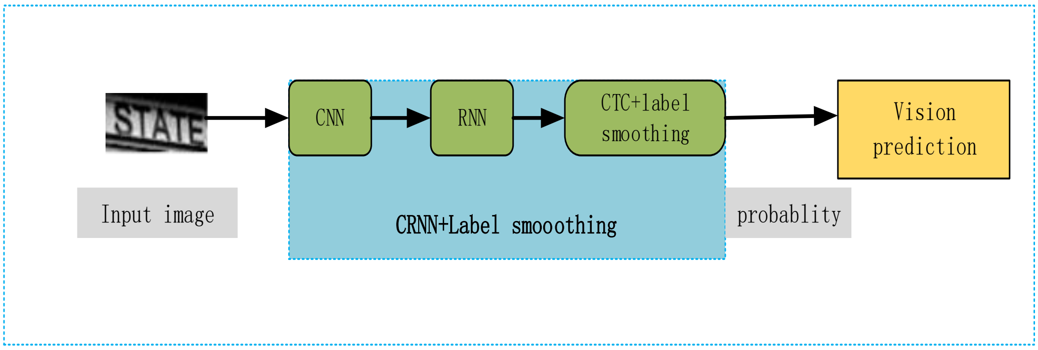

Figure 4. Schematic diagram of the CRNN+label smoothing model.

3.2. Language Model

CTC is a classification algorithm that solves temporal data, where training involves obtaining labels for each frame of data. Models using the CTC criterion as a loss function are end-to-end model training and do not require pre-alignment of data; only input and output sequences are required for training. The main drawbacks of the CTC model are that it still has the assumption of conditional independence between data, and that the CTC model only has the ability to model acoustically and lacks some language model capabilities [24]. Enhanced language modeling usually requires the acquisition of linguistic features, which refers to inferring the class of a character by considering the context between characters, not based on the glyphic features of the character, and is usually paired with visual features extracted by visual models, The language model BCN is proposed in ABINet [25] because the purpose of the language model in ABINet [25] is to iteratively correct and check letters by fusing visual and language model features and iteratively checking n times. Considering the actual running time and the limited semantic information extracted after the iterative iterations, only the language model BCN is added to the CRNN network after the feature fusion iterations without adding the fused iterative language model. From Figure 5 it can be seen that the output of the CRNN is used as input to the language model to provide a priori knowledge for the language model to carry out the acquisition of semantic information, and the gradient is back-propagated as an auxiliary task for CRNN recognition so that the output of the CRNN contains more semantic information.

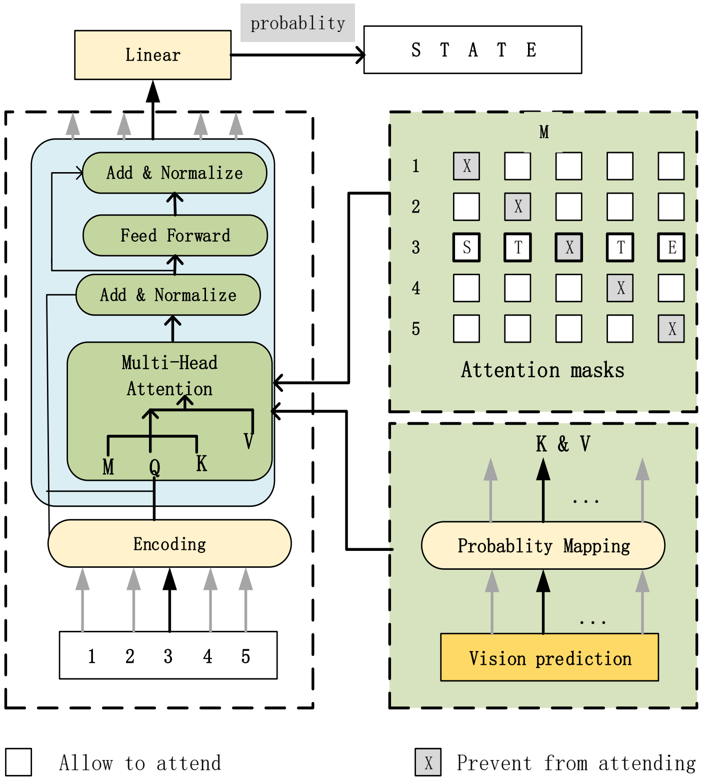

Figure 5. Language model BCN.

The prediction results of the visual model are used to provide a priori knowledge for the language model, and text recognition is performed at both visual and contextual semantic levels to obtain more comprehensive information about the text. Each layer of the BCN is a series of multi-head attention feed forward networks, residual connections, and layer normalization. The network takes as input a sequential encoding of character positions as a non-character probability vector, and the character probability vector is passed directly into the multi-head attention module. In the multi-head attention mechanism, the diagonal attention mask is designed to avoid seeing the current character and to achieve simultaneous access to information to the left and right of the character, combining the left and right information to make a prediction simultaneously.

$$M_{ij} = \begin{cases} 0, & i \neq j \\ \infty, & i = j' \end{cases} \tag{3}$$

$$K_i = V_I = P(y_i)W_l, \tag{4}$$

$$F_{mha} = softmax\left(\frac{QK^T}{\sqrt{C}} + M\right)V, \tag{5}$$

where, from Equations (3)–(5), given a text string $y = (y_1, \cdots\cdots, y_n)$ with text length n and category c, C is the feature size, $Q \in R^{T\times C}$ is the position encoding of the character order, the position encoding of the first layer of character order, and the others are the output of the last layer, and T is the length of the character sequence. K, $V \in R^{T\times C}$ is obtained from the character probability $P(y_i) \in R^c$, $W_l \in R^{c\times C}$ is the linear mapping matrix, and $M \in R^{T\times T}$ is the attention mask matrix, blocking attention to the current position, after stacking the BCN layers into a depth architecture, determining the bidirectional expression, F_l, of the text, y.

The BCN does not use self-attention to avoid information leakage across steps, and each time step of the BCN is computed independently and in parallel, which is efficient. The schematic diagram of adding the language model to CRNN is shown in Figure 6. Considering Figures 5 and 6 together, it can be visualized that the prediction results from the visual model are input to the language model after masking, multi-headed attention, feed-forward network, and normalization to obtain the final prediction result STATE.

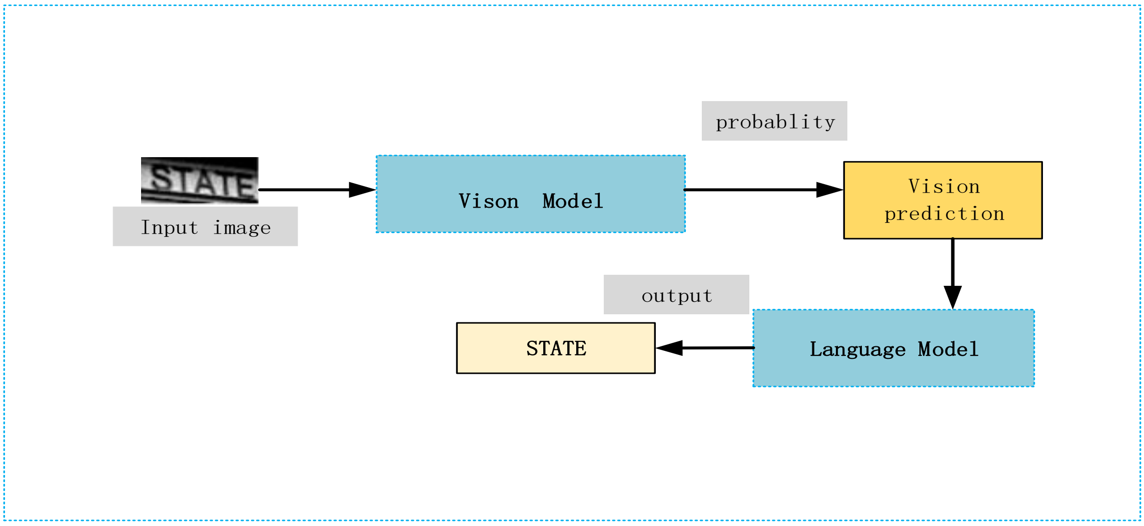

Figure 6. Schematic diagram of CRNN+BCN.

4. Experiments

4.1. Datasets

In the field of text recognition, the demand for data volume is very large. Compared with the training data volume of a few thousand or a few hundred in the field of text detection, the training data volume in the field of text recognition is in the millions. The datasets used in this paper are all public datasets, which can be divided into synthetic datasets and real datasets. Since the training of network models requires a large amount of data as support, most of the text recognition models are trained using synthetic datasets; real datasets are used for the evaluation of the training results of text recognition models.

① Synthetic dataset for training

MJSynth [26] is a synthetic plain English text dataset. The dataset contains 3000 folders with about 9 million images, rendering the text onto natural images and then performing a random transformation. The words in each image in the dataset are labeled and the character orientation is mainly horizontal. SynthText [27] is also a synthetic text dataset, but the difference is that SynthText is designed mainly for text detection, so the text is rendered onto the complete natural image, consisting of 800,000 scene images. To accommodate text recognition, the words are cropped according to the word annotation bounding boxes in the experiments, and a total of about 7 million text images are cropped.

② Real datasets for evaluation

Depending on the format of the text in the image, the real dataset can be divided into regular and irregular text. Regular datasets include IIIT5k-Words (IIIT5k), SVT (Street View Text), and IC13 (ICDAR2013). Irregular datasets include IC15 (ICDAR2015), SVTP (Street View Text Perspective), and CUTE80.

IIIT5k-Words (IIIT5k) [28] consists of 3000 test images collected from the Internet. The text in the images is mostly regular text, and, for each image, two dictionaries of different sizes, 1000 words and 50 words, are matched. Each dictionary consists of real annotations and other commonly used words; the images in SVT [29] are mainly cropped from 249 Google Street View views and contain 647 low-resolution and low-noise text images, most of which are horizontal text images. Each image matches a 50-word dictionary; the vast majority of text images in ICDAR 2013 (IC13) [30] are from IC03, with new images added for data augmentation, for a total of 1015 text images, most of which are regular text images, with some text images blurred due to uneven illumination.

ICDAR 2015 [31] from Challenge 4 of the ICDAR 2015 Robust Reading Competition, called incidental scene text, consists mainly of plain English text of multi-directional scenes. This dataset consists of some randomly taken street view images with low resolution, and most of the text in the figures is relatively small and blurred, so it is relatively difficult to detect. ICDAR 2015 divides the training and test sets into 1000 and 500 images, each of which contains multi-directional text, which is labeled in word units using a rectangular box of 4 points; SVTP [32] was selected from the side view of Google Street View. The dataset consists of 645 cropped text images, most of which have distortion factors such as low resolution, noise, and blur. Each image provides a dictionary of 50 words. CUTE80 (CUTE) [33] contains 288 high-resolution text images cropped from the original dataset. The text in this dataset is mainly curved and directional text, and no associated dictionaries are provided.

In this paper, we are limited by the experimental equipment and the actual running time, so we use the synthetic dataset MJ in the training phase, and then we can add the dataset ST for training if the experimental conditions allow.

4.2. Implementation Details

In this paper, we use public datasets in the field of text recognition for training and testing. The experimental platform is a Linux system, GPU: RTX 3090 (24 GB)*3, CPU 45Vcpu AMD EPYC 7543 32-Core Processor, memory 240 GB. The deep learning tool is Pytorch.

To verify the effectiveness of text recognition algorithms, evaluation metrics usually use a character recognition rate or word recognition rate, both of which compare the predicted value with the real value. The character recognition rate is calculated in character units, counting the ratio of the number of correctly recognized characters to the total number of actual characters. The word recognition rate is calculated in word units, counting the ratio of the number of correctly recognized words to the total number of actual words in the image. Word recognition rates are more demanding than character recognition rates. In the process of character recognition, a single character error can be tolerated; however, when a word recognition rate is used, each character in a word must be recognized correctly. The above two evaluation metrics are commonly used in English datasets. In this paper, word recognition accuracy is used as an evaluation index for the merits of text recognition algorithms, which is calculated as in Equation (6):

$$\text{Accuracy} = \frac{N}{M} \times 100\%, \tag{6}$$

In Equation (6), M represents the number of all recognized characters, and N represents the number of samples with completely correct recognition.

4.2.1. Comparison of Experimental Results between the Actual Run Baseline Model and the Original Baseline Model

In order to evaluate the text recognition performance of the improved CRNN network model objectively, considering that the test data in the original CRNN paper did not include all six datasets, the CRNN test data involved in the comparison are from the replication of Benchmark [16], and the CRNN-base represents the results obtained from the actual experiments on the Linux experimental platform. From Table 1, we can see that the CRNN-base has 0.05%, 3.83%, 3.01%, 6.53%, 5.07%, and 0.75% growth over CRNN in IC13, SVT, IIIT5K, IC15, SVTP, and CUTE, respectively, and the experimental data are real and reliable.

Table 1. Comparison table of CRNN baseline experiments (accuracy of English dataset (%)).

Methods	IC13	SVT	IIIT5K	IC15	SVTP	CUTE
CRNN	91.10	81.60	82.90	69.40	70.00	65.50
CRNN-base	91.15	85.43	85.91	75.93	75.07	66.25

4.2.2. Comparison of Adding Label Smoothing with the Baseline Model Approach

The experimental comparison table after adding label smoothing is shown in Table 2. Considering that label smoothing is mostly applied in the classification task and the default value of α is 0.1, the optimal value of α can be taken as 0.01 when adding label smoothing in other tasks. Although text recognition can also be considered a multi-classification problem, the applicability of label smoothing in the field of text recognition for the value of α needs to be considered. The optimal result of the label smoothing parameter, α, needs to be determined, and this paper uses a GridSearch method to verify it around 0.01 versus 0.1. Because the GridSearch is a means of tuning the parameters by the exhaustive search method, in all the candidates of the parameter selection, by circular traversal, trying every possibility, the best performing parameter is the final result of the selection. The specific experimental results are shown in Table 2.

Table 2. Comparison table of α values (accuracy of English dataset (%), in the Table 2 "+" indicates an increase over CRNN-base and "−" indicates a decrease under CRNN-base).

Methods	IC13	SVT	IIIT5K	IC15	SVTP	CUTE
CRNN-base	91.15	85.43	85.91	75.93	75.07	66.25
LS (α = 0.005)	94.79 (+3.64)	90.44 (+5.01)	89.61 (+3.70)	82.15 (+6.22)	80.70 (+5.63)	73.40 (+7.15)
LS (α = 0.01)	93.42 (+2.27)	89.30 (+3.87)	89.27 (+3.36)	82.28 (+6.35)	80.04 (+4.97)	72.40 (+6.15)
LS (α = 0.05)	94.12 (+2.97)	90.12 (+4.69)	90.56 (+4.65)	81.70 (+5.77)	79.22 (+4.15)	73.34 (+7.09)
LS (α = 0.1)	89.71 (−1.44)	83.30 (−2.13)	83.91 (−2.00)	71.98 (−3.95)	69.61 (−5.46)	60.48 (−5.77)

From the experimental results of GridSearch, it can be seen that, when the default value of 0.1 is used for label smoothing, there is a decrease in the accuracy rate based on the CRNN-base, indicating that the parameter default value is not ideal in the field of text recognition; therefore, the algorithms and modules in other fields, when applied to new fields, require parameter adjustment to find the best parameters. From the experimental results, it can be seen that, except for an α default value of 0.1, other values have different degrees of growth. Especially, the best effect is achieved when α = 0.005, where the growth points on the irregular dataset are more than those on the regular dataset, which indicates that the applicability is stronger in irregular text, and in IC13, SVT, IIIT5K, IC15, SVTP, and CUTE, the CRNN-base models grow 3.64%, 5.01%, 3.70%, 6.22%, 5.63% and 7.15%, respectively, which is significant. As a result, the field of label smoothing applications has been extended from classification, machine translation, image segmentation, and speech recognition to text recognition. By implementing regularization, the model is prevented from predicting labels too confidently during training to improve generalization ability, and can also be tuned with parameters to achieve significant improvements in recognition accuracy.

4.2.3. Comparison of Adding Language Model with the Baseline Model Approach

The experimental comparison table after adding the BCN language model is shown in Table 3. From the experimental results, it can be seen that adding the language model to extract semantic information and adding label smoothing can achieve the effect of improving the accuracy rate, and there are different degrees of improvement in both regular and irregular texts, with more growth points in the irregular datasets than in the regular datasets. In IC13, SVT, IIIT5K, IC15, SVTP, and CUTE, the increases over the base model are 2.35%, 3.24%, 2.62%, 4.71%, 4.39%, and 5.21%, respectively. From the experimental results, it can be seen that there is a problem of incomplete information acquisition when considering the text information from the visual model alone for recognition. The addition of the language model not only considers the visual aspect of a single character’s glyphic features but also infers the category of the character from the context between characters, broadening the access to information and improving the overall effect.

Table 3. Comparison table after adding the language model (accuracy of English dataset (%)).

Methods	IC13	SVT	IIIT5K	IC15	SVTP	CUTE
CRNN-base	91.15	85.43	85.91	75.93	75.07	66.25
CRNN+BCN	93.50 (+2.35)	88.67 (+3.24)	88.53 (+2.62)	80.64 (+4.71)	79.46 (+4.39)	71.46 (+5.21)

4.2.4. Comparison of Ablation Experiments

To understand the effect of label smoothing and the language model better, the experimental effects of adding label smoothing and the language model are compared with adding label smoothing alone, adding the language model alone, and the baseline model, respectively.

As seen in the comparison table of ablation experiments in Table 4, compared with the CRNN-base, the addition of label smoothing (α = 0.005) and the language model BCN resulted in increases of 4.00%, 5.27%, 5.22%, 7.77%, 7.37%, and 9.03% in IC13, SVT, IIIT5K, IC15, SVTP and CUTE, respectively. Compared with smoothing with the label addition alone (α = 0.005), the addition of label smoothing (α = 0.005) and the language model BCN resulted in increases of 0.36%, 0.26%, 1.52%, 1.55%, 1.74%, and 1.88% in IC13, SVT, IIIT5K, IC15, SVTP, and CUTE, respectively. Compared with the language model BCN alone, the addition of label smoothing (α = 0.005) and the language model BCN resulted in increases of 1.63%, 2.03%, 2.60%, 3.06%, 2.98%, and 3.82% in IC13, SVT, IIIT5K, IC15, SVTP, and CUTE, respectively. Comparing the growth of label smoothing alone with the growth of the language model alone, it can be seen intuitively that the growth points of adding language model accuracy are much higher than the growth points of label smoothing, and the result of fusing label smoothing and the language model is higher than the simple sum of the growth points of label smoothing and the language model, which proves the effectiveness of verifying the effectiveness of label smoothing and the language model on text recognition, with the highest growth point number reaching 9.03%. The overall experimental effect is improved significantly.

Table 4. Comparison table of ablation experiments (accuracy of English dataset (%)).

Methods	IC13	SVT	IIIT5K	IC15	SVTP	CUTE
CRNN-base	91.15	85.43	85.91	75.93	75.07	66.25
LS (α = 0.005)	94.79	90.44	89.61	82.15	80.70	73.40
CRNN+BCN	93.50	88.67	88.53	80.64	79.46	71.46
CRNN+BCN+LS (α = 0.005)	95.15	90.70	91.13	83.70	82.44	75.28

4.2.5. Comparison of the Improved CRNN Model with Other Methods

The results in the comparison table in Table 5 show that the improved CRNN network, on the first five evaluation datasets, significantly outperforms the models RARE [8], R2AM [9], STAR-Net [10], GRCNN [11], Rosetta [12], Benchmark [13], SEED [14], and ViTSTR [15] in recognition performance. The models TRBA [16], Cascade Attention [17], Text is Text [18], and Character-Context Decoupling [19] outperform the improved CRNN network effect on some datasets, this is indicated by highlighting the results in bold in Table 5; although there is a gap, the gap is not obvious, and the main gap is in the sixth dataset, CUTE. Analyzing the reasons for the gap: compared with other low-resolution, multi-directional irregular text datasets, the CUTE dataset has mainly curved text and directional text, and no associated lexicon is provided, while other datasets, although there are directional and curved characteristics, provide an associated lexicon. Therefore, the gap in recognition effect in different datasets in this paper is related to the lexicon, and the comparison with other methods in the table further demonstrates the superiority of the method in this paper. In the process of experimental comparison, the degree of improvement of the effect of the irregular dataset is greater than that of the regular text, but, from the overall recognition accuracy, it can be seen that the recognition of regular text has reached more than 90%, while the recognition accuracy of irregular text has not yet reached 90%. So the network in the recognition of irregular text should be further improved.

Table 5. Comparison table of methods (accuracy of English dataset (%)).

Methods	IC13	SVT	IIIT5K	IC15	SVTP	CUTE
CRNN-base	91.15	85.43	85.91	75.93	75.07	66.25
RARE [8]	92.60	85.80	86.20	74.50	76.20	70.40
R2AM [9]	90.20	82.40	83.40	68.90	72.10	64.90
STAR-Net [10]	92.80	86.90	87.00	76.10	77.50	71.70
GRCNN [11]	90.90	83.70	84.20	71.40	73.60	68.10
Rosetta [12]	90.90	84.70	84.30	71.20	73.80	69.20
Benchmark [13]	93.60	87.50	87.90	77.60	79.20	74.00
SEED [14]	92.80	89.60	93.80	80.00	81.40	**83.60**
ViTSTR [15]	92.40	87.70	88.40	78.50	81.80	**81.30**
TRBA [16]	93.10	88.90	**92.10**	-	79.50	**78.20**
Cascade Attention [17]	**96.80**	89.50	90.30	-	78.50	**78.90**
Text is Text [18]	93.30	89.90	**92.30**	76.90	**84.40**	**86.30**
Character Context Decoupling [19]	92.21	85.93	**91.90**	-	-	**83.68**
CRNN+BCN+LS (our)	95.15	90.70	91.13	83.70	82.44	75.28

5. Conclusions

In this paper, we propose an improved CRNN for scene text recognition. The improved CRNN model is used for sequence recognition of text in images, trained on synthetic datasets, and tested on six public datasets. Experimental results show that the improved CRNN accuracy outperforms the original CRNN network and is compared with other experimental methods with good results. The main contribution of this paper is to improve the original CRNN network and propose a new idea of text recognition based on a neural network. The CRNN network model retains the overall architecture of the original network, and the use of label smoothing in predicting output results can effectively improve the generalization ability of the model, improve the anti-interference ability of the model, prevent the generation of over-fitting, and thus improve the recognition accuracy. The smoothing loss function in speech recognition is introduced into the field of text recognition, and the CTC loss function is redefined. The smoothing loss function in speech recognition is introduced into the field of text recognition, and the CTC loss function is redefined. A language model is added to fuse sequence information with language information for text recognition, increasing the information acquisition channels and ultimately achieving the goal of improving the accuracy of text recognition. With the addition of the language model, the overall recognition accuracy is improved, but the number of parameters also increases and the network run time becomes longer. Therefore, network optimization for the improved network model is needed in future work to reduce the number of parameters and running time. In addition, the public dataset is mainly in English, and CRNN has achieved certain results in both Chinese and English recognition, but real-life text images can be mixed with multiple languages, and further research is needed for the recognition of mixed text.

Author Contributions: Conceptualization, W.Y.; methodology, W.Y.; software, W.Y.; validation, W.Y. and M.I.; formal analysis, W.Y.; investigation, W.Y., A.H. and M.I.; resource, A.H. and M.I.; data curation, W.Y.; writing—original draft preparation, W.Y.; writing—review and editing, M.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the Natural Science Foundation of China (62166043, U2003207).

Data Availability Statement: Publicly available datasets were analyzed in this study. Our datasets. can be obtained from [https://github.com/clovaai/deep-text-recognition-benchmark] (11 December 2020), and [https://github.com/FangShancheng/ABINet] (10 June 2021).

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

References

1. Liu, C.; Chen, X.; Luo, C.; Jin, L.; Xue, Y.; Liu, Y. A deep learning approach for natural scene text detection and recognition. *Chin. J. Graph.* **2021**, *26*, 1330–1367. [CrossRef]
2. 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. *arXiv* **2015**, arXiv:1507.05717. [CrossRef] [PubMed]
3. 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, Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376.
4. Liu, Y.; Wang, Y.; Shi, H. A Convolutional Recurrent Neural-Network-Based Machine Learning for Scene Text Recognition Application. *Symmetry* **2023**, *15*, 849. [CrossRef]
5. Lei, Z.; Zhao, S.; Song, H.; Shen, J. Scene text recognition using residual convolutional recurrent neural network. *Mach. Vis. Appl.* **2018**, *29*, 861–871. [CrossRef]
6. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
7. Graves, A.; Mohamed, A.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649.
8. Shi, B.; Wang, X.; Lyu, P.; Yao, C.; Bai, X. Robust scene text recognition with automatic rectification. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4168–4176.
9. Lee, C.-Y.; Osindero, S. Recursive recurrent nets with attention modeling for OCR in the wild. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2231–2239.
10. Liu, W.; Chen, C.; Wong, K.-Y.K.; Su, Z.; Han, J. Star-net: A spatial attention residue network for scene text recognition. *BMVC* **2016**, *2*, 7.
11. Wang, J.; Hu, X. Gated recurrent convolution neural network for OCR. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 334–343.
12. Borisyuk, F.; Gordo, A.; Sivakumar, V. Rosetta: Large scale system for text detection and recognition in images. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 71–79.
13. Baek, J.; Kim, G.; Lee, J.; Park, S.; Han, D.; Yun, S.; Oh, S.J.; Lee, H. What is wrong with scene text recognition model comparisons? Dataset and model analysis. In Proceedings of the 2019 IEEE/CVF international Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4715–4723.

14. Qiao, Z.; Zhou, Y.; Yang, D.; Zhou, Y.; Wang, W. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13528–13537.
15. Atienza, R. Vision transformer for fast and efficient scene text recognition. In Proceedings of the Document Analysis and Recognition—ICDAR 2021: 16th International Conference, Lausanne, Switzerland, 5–10 September 2021; Proceedings, Part I 16. Springer International Publishing: Cham, Switzerland, 2021; pp. 319–334.
16. Baek, J.; Matsui, Y.; Aizawa, K. What if we only use real datasets for scene text recognition? Toward scene text recognition with fewer labels. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3113–3122.
17. Zhang, M.; Ma, M.; Wang, P. Scene text recognition with cascade attention network. In Proceedings of the 2021 International Conference on Multimedia Retrieval, New York, NY, USA, 21–24 August 2021; pp. 385–393.
18. Bhunia, A.K.; Sain, A.; Chowdhury, P.N.; Song, Y.-Z. Text is text, no matter what: Unifying text recognition using knowledge distillation. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 983–992.
19. Liu, C.; Yang, C.; Yin, X.C. Open-Set Text Recognition via Character-Context Decoupling. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4523–4532.
20. Liu, M.; Zhou, L. A cervical cell classification method based on migration learning and label smoothing strategy. *Mod. Comput.* **2022**, *28*, 1–9+32.
21. Müller, R.; Kornblith, S.; Hinton, G.E. When does label smoothing help? *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 422.
22. Zhao, L. Research on User Behavior Recognition Based on CNN and LSTM. Master's Thesis, Nanjing University of Information Engineering, Nanjing, China, 2021. [CrossRef]
23. Kim, S.; Seltzer, M.L.; Li, J.; Zhao, R. Improved training for online end-to-end speech recognition systems. *arXiv* **2017**, arXiv:1711.02212.
24. Qin, C. Research on End-to-End Speech Recognition Technology. Ph.D. Thesis, Strategic Support Force Information Engineering University, Zhengzhou, China, 2020. [CrossRef]
25. Fang, S.; Xie, H.; Wang, Y.; Mao, Z.; Zhang, Y. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7098–7107.
26. Cheng, L.; Yan, J.; Chen, M.; Lu, Y.; Li, Y.; Hu, L. A multi-scale deformable convolution network model for text recognition. In Proceedings of the Thirteenth International Conference on Graphics and Image Processing (ICGIP 2021); SPIE: Paris, France, 2022; Volume 12083, pp. 627–635.
27. Gupta, A.; Vedaldi, A.; Zisserman, A. Synthetic data for text localisation in natural images. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2315–2324.
28. Mishra, A.; Alahari, K.; Jawahar, C.V. Top-down and bottom-up cues for scene text recognition. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2687–2694.
29. Wang, K.; Babenko, B.; Belongie, S. End-to-end scene text recognition. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1457–1464.
30. Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; Bigorda, L.G.; Mestre, S.R.; Mas, J.; Mota, D.F.; Almazan, J.A.; de las Heras, L.P. ICDAR 2013 robust reading competition. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 1484–1493.
31. Phan, T.Q.; Shivakumara, P.; Tian, S.; Tan, C.L. Recognizing text with perspective distortion in natural scenes. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 569–576.
32. 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. [CrossRef]
33. Cheng, Z.; Xu, Y.; Bai, F.; Niu, Y.; Pu, S.; Zhou, S. Aon: Towards arbitrarily-oriented text recognition. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5571–5579.

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.