*mathematics*

MDPI

*Article*

# Deep-Learning-Based Complex Scene Text Detection Algorithm for Architectural Images

**Weiwei Sun [1], Huiqian Wang [1], Yi Lu [1], Jiasai Luo [1], Ting Liu [1], Jinzhao Lin [1], Yu Pang [1,*] and Guo Zhang [1,2,*]**

[1] Chongqing Key Laboratory of Photoelectronic Information Sensing and Transmitting Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[2] School of Medical Information and Engineering, Southwest Medical University, Luzhou 646000, China
* Correspondence: pangyu@cqupt.edu.cn (Y.P.); zhangguo@swmu.edu.cn (G.Z.); Tel.: +86-13372653576 (Y.P.); +86-13438598188 (G.Z.)

**Abstract:** With the advent of smart cities, the text information in an image can be accurately located and recognized, and then applied to the fields of instant translation, image retrieval, card surface information recognition, and license plate recognition. Thus, people's lives and work will become more convenient and comfortable. Owing to the varied orientations, angles, and shapes of text, identifying textual features from images is challenging. Therefore, we propose an improved EAST detector algorithm for detecting and recognizing slanted text in images. The proposed algorithm uses reinforcement learning to train a recurrent neural network controller. The optimal fully convolutional neural network structure is selected, and multi-scale features of text are extracted. After importing this information into the output module, the Generalized Intersection over Union algorithm is used to enhance the regression effect of the text bounding box. Next, the loss function is adjusted to ensure a balance between positive and negative sample classes before outputting the improved text detection results. Experimental results indicate that the proposed algorithm can address the problem of category homogenization and improve the low recall rate in target detection. When compared with other image detection algorithms, the proposed algorithm can better identify slanted text in natural scene images. Finally, its ability to recognize text in complex environments is also excellent.

**Keywords:** building detection; geographic position; EAST; smart cities

**MSC:** 68T07; 68U15



**Citation:** Sun, W.; Wang, H.; Lu, Y.; Luo, J.; Liu, T.; Lin, J.; Pang, Y.; Zhang, G. Deep-Learning-Based Complex Scene Text Detection Algorithm for Architectural Images. *Mathematics* **2022**, *10*, 3914. https://doi.org/10.3390/math10203914

Academic Editor: Teng Li

Received: 12 September 2022
Accepted: 14 October 2022
Published: 21 October 2022



## 1. Introduction

In recent years, more and more devices (smartphones, smart watches, high-definition surveillance cameras, etc.) for acquiring images and videos have been widely used in various industries. People have access to massive amounts of image data. Some important text information is often included in these image data, such as license plate numbers, product introduction text in billboards, road information and direction indication text in street signs, etc. [1,2]. Therefore, the text in the image is detected and recognized by the computer, and the obtained text information plays an important role in promoting the development of human–computer interaction [3], geographic location positioning [4], real-time translation [5], robot navigation [6], and industrial automation [7]. However, the texts in the images have different sizes, different font shapes and orientations, and even overlap and contamination, which make text detection very difficult. Therefore, text extraction in natural scenes has gradually become a research hotspot in the field of image processing.

The traditional text-detection algorithms are cumbersome and less robust. For example, Neumann et al. [8] searched for candidate character features through the Maximally Stable Extremal Region algorithm, and combined the extracted features into word or text regions according to custom rules or classifiers. This method is relatively efficient, but the

performance is poor in the case of uneven lighting. Louloudis et al. [9] used the parallel or symmetrical properties of upper and lower edges between text lines to achieve the effective detection of text candidates. However, this method lags behind the deep learning methods that have appeared in recent years in terms of accuracy and adaptability. In particular, dealing with scene features such as low resolution and geometric distortion, the text in the picture is disturbed by the complex background, which increases the difficulty of text detection.

With the continuous progress of deep learning [10–18] technology and the popularization of high-computing-power hardware in recent years, the application fields of deep learning are becoming more and more extensive, such as computer vision (biometric recognition, image processing, video analysis), natural language processing (speech recognition, text data mining, text translation), data mining (consumption habit, weather data, recommendation systems), and composite applications (unmanned driving, unmanned aerial vehicles, robots), etc. Text detection algorithms based on deep learning have become the mainstream direction of current text detection technology research. The basic objective of a text detection algorithm is to use a neural network structure to automatically extract text features from natural scene images. The weight parameters are updated based on feedback via a loss function to achieve text localization [19]. Natural scene text detection methods based on deep learning can be divided into three types: bounding box (BBOX) regression [20–24], semantic segmentation [25–28], and a combination of these two [29–33]. Of these, the text detection method based on BBOX regression is the most widely used one. A combination of semantic segmentation and BBOX regression can achieve better text detection results. However, such a method uses more steps, which increases the processing time. The operation of the combined method is similar to that of the Mask-RCNN algorithm [34]. Huang et al. [35] used the MSER algorithm to find candidate characters, and then used a deep neural network algorithm as a classifier to screen out the final text lines. Jaderberg et al. [36] scanned images with the help of sliding windows and used a convolutional neural network model to generate multi-scale feature images. Shi et al. [37] proposed a Connectionist Text proposal network model to extract features by combining a CNN and RNN deep network. This method enhances the connection of text lines and improves the detection accuracy. However, this method can only detect horizontal text. In addition, the receptive field range of common convolutional neural networks is limited, and it is more challenging to directly detect long text lines. Therefore, Zhi et al. [38] proposed the SegLink text detection algorithm. The method detects the local areas of words or text lines, and then connects these local areas to form complete words or text lines. However, the subsequent processing method is complex and slow, and the detection of text with long feeling fields is not very good. In addition, Zhou et al. [39] proposed an efficient and accurate scene text detector (EAST) algorithm in 2017. The algorithm can predict multi-angle quad text areas in natural scene images. The previous text detection algorithm often contained intermediate steps such as candidate text box proposal, text box formation, word segmentation, and related post-processing, which made the algorithm structure more complicated. As an end-to-end text detection algorithm, EAST simplifies the entire work process into two phases. First, the full convolutional network is used to obtain multi-scale characteristic images, and then the feature fusion is carried out to obtain a feature map; the position information of the text box is predicted on this feature image. Then, non-maximum suppression and fusion of text boxes are performed; finally, the predicted text box is output. The EAST algorithm has a simple structure and good performance. The output text box is also suitable for the detection of the text area in the street sign scene. However, the EAST algorithm has a poor effect on complex scenes and in multi-scale text detection (Figure 1).

**Figure 1.** Textual information in multi-scale and complex scene natural imagery.

In this study, we proposed a Neural Architecture Search—Feature Pyramid Network EAST (NEAST) text detection algorithm to solve the problem whereby images are difficult to detect in complex scenes and multi-scale text. The algorithm consists of a feature extraction layer and an output layer. The feature extraction layer adopts an automatic architecture search network. It can extract features at various levels with diverse scales. The output layer first addresses the class imbalance problem in the scene image, and then improves the Intersection over Union (IoU) [40] algorithm to obtain the Generalized Intersection over Union (GIoU) [41] algorithm. A text suggestion box is selected to obtain the final text detection result. The algorithm not only better solves the classification homogeneity problem, but also improves the problem of low recall in target detection. Compared with other image detection algorithms, our proposed NEAST algorithm can better recognize skewed text in natural scene images. The algorithm shows excellent text recognition capability in complex environments.

## 2. Materials and Methods

### *2.1. EAST Algorithm*

Scene text detection methods have achieved encouraging results on various benchmarks. However, these methods, even those using deep neural network models, have shortcomings when dealing with challenging scenarios. Because the overall performance of a text detection model depends on the interactions among the modules in the algorithmic model, a simple model can optimize the loss function and neural network structure in a targeted manner and improve text detection. Therefore, text regions in natural scene images can be detected quickly and accurately using a simple and efficient EAST algorithm. The model structure is depicted in Figure 2. The simple structure and fast operation are the advantages of the EAST algorithm. However, its accuracy of text detection in complex scenes is not satisfactory.

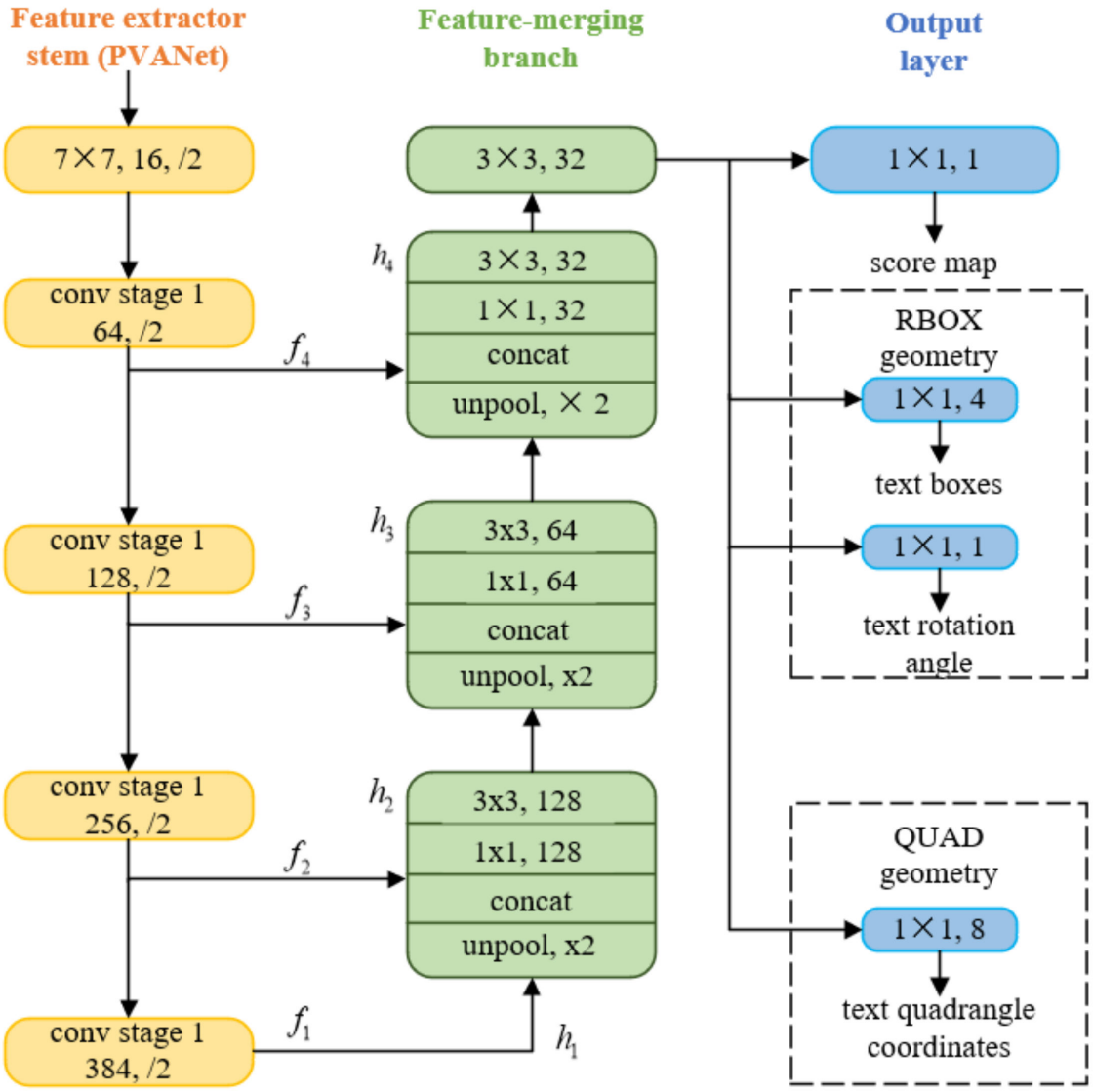



**Figure 2.** EAST algorithm model.

The network structure of EAST is shown in Figure 2. We use the Performance vs. Accuracy Network (PVANet) to extract features, and merge the features of different layers after upsampling. Then, the final score and box are predicted. The box is represented by RBOX and QUAD. If box data are annotated in the form of RBOX, the model finally predicts the 1-channel score_map and the 4-channel box_map. If the box data are annotated in the form of QUAD, the model finally predicts the 1-channel score_map and the 8-channel BOX_map.

#### 2.1.1. Feature Extraction Network

After an image is input, feature extraction is performed using a feature pyramid network (FPN). Pre-training is performed using a CNN with interleaved convolutional and pooling layers. Four levels of feature maps, $f_i$, are obtained, whose sizes are 1/32, 1/16, 1/8, and 1/4 of the original image. The objective of this step is to address the problem of scale diversity of text lines. Low- and high-level features are used to predict smaller- and larger-sized text lines, respectively.

2.1.2. Feature Fusion

In the feature fusion step, the feature map extracted in the preceding stage is first de-pooled. The size of the feature map is increased, and it is connected in series with the current feature map. Next, a $1 \times 1$ convolution kernel is used to perform the convolution operation to reduce the calculation load of the model. Subsequently, the convolution operation is performed through a $3 \times 3$ convolution kernel to fuse the feature information. Finally, the result of the text detection feature fusion stage is output. When the feature fusion process is completed, a $3 \times 3$ convolution kernel is used to perform the convolution operation. The resulting final feature map is input to the output layer. The feature fusion operates as follows:

$$g_i = \begin{cases} \text{unpool}(h_i) & \text{if } i \leq 3 \\ \text{conv}_{3\times3}(h_i) \text{ if } i = 4 & \text{if } i = 1 \end{cases}$$
$$h_i = \begin{cases} f_i & \text{otherwise} \\ \text{conv}_{3\times3}(\text{conv}_{1\times1}([g_{i-1}; f_i])) & \text{othere} \end{cases} \quad (1)$$

2.1.3. Output Layer

The output layer is obtained using the convolution operation with many $1\times1$ convolution kernels. Then, 32-channel feature maps are projected to generate 1-channel fractional and multi-channel geometry feature maps. The geometry output can be rectangular or quadrilateral (see Table 1).

**Table 1.** EAST algorithm output.

| Geometry | Channels | Description |
|---|---|---|
| AABB | 4 | $G = R = \{d_i \mid i \in \{1,2,3,4\}\}$ |
| RBOX | 5 | $G = \{R, \theta\}$ |
| QUAD | 8 | $G = Q = \{(\Delta x_i, \Delta y_i) \mid i \in \{1,2,3,4\}\}$ |

In Table 1, AABB is a horizontal rectangular frame. The four channels represent the four distances from the pixel positions to the top, right, bottom, and left borders of the rectangle, respectively. RBOX is a rotating rectangular box, and the geometric shape is represented by the horizontal BBOX of four channels and the rotation angle $\theta$ of one channel. QUAD is an arbitrary quadrilateral, and numeral 8 represents the coordinate offsets from the four vertices of the rectangle to the pixel position. Because each coordinate offset includes two values of the abscissa and ordinate $(\Delta x_i, \Delta y_i)$, the output of its geometry needs to contain eight channels.

This method mainly includes a fully convolutional network and non-maximum suppression. The algorithm model can flexibly generate word-level or text-line-level text prediction boxes, and the predicted geometric shapes can be rotated or horizontal boxes. However, the picture detection in complex scenes is not satisfactory.

*2.2. NEAST Oblique Text Detection Method*

The objective of the proposed algorithm is to obtain a network structure in the search space and set it as a sub-network through a recurrent neural network (RNN) [42] controller. Subsequently, this network structure is trained on a dataset. After its accuracy R is obtained by testing on the validation set, the accuracy values are returned to the controller. The controller continues to optimize to obtain another network structure. This is repeated until an optimal feature extraction network structure is obtained. Thereafter, the output layer is improved to enhance the accuracy, and the final detection result is obtained.

The proposed detection model is shown in Figure 3. We used the Learning Scalable Feature Pyramid Architecture for Object Detection (NAS-FPN) to combine the features of multiple layers. NAS-FPN reorganizes the feature images on five scales. FPN uses five

layers of resolution features (C3, C4, C5, C6, C7), and C5 is subsampled to obtain C6 and C7. The resolution downsampling of the features in the five layers is (8, 16, 32, 64, 128), and (P3, P4, P5, P6, P7) can be obtained through FPN. The output layer convolves the extracted feature map through several $1 \times 1$ convolution kernels to generate the score plot and the rectangle detection block diagram in any direction. The text box regression is guided by the GIoU algorithm. Positive and negative samples are selected from the prior box. Then, the positive samples are encoded according to the proposed encoding. The problem that IoU cannot be optimized without overlap is solved, and it can also be used as a measure in the target detection task. The regression effect of the text box will be improved. We use the focus loss (FL) function (including position loss and classification loss) to solve the problem of positive and negative sample imbalance and hard classification sample learning.

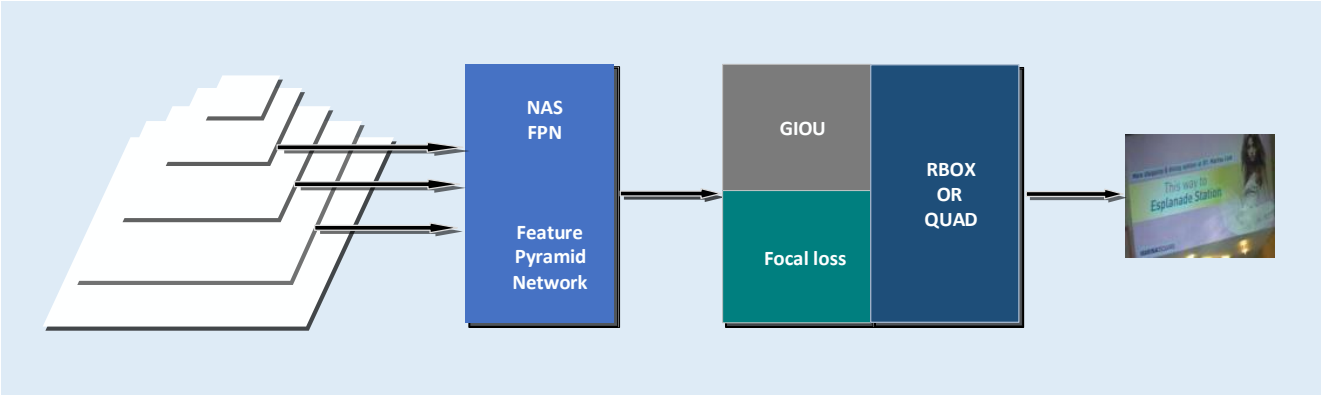



**Figure 3.** Text detection model based on NEAST.

#### 2.2.1. Feature Extraction Module

Text regions in natural scene images are multi-scale and multi-object. Two types of features (low resolution, strong semantics; high resolution, low semantics) are combined to form the FPN structure through top-down paths and lateral connections. This structure is an inevitable choice for feature extraction. The text region occupies a small proportion of the image area. The shape of the region is mostly narrow and long. The target size and structure are significantly different from other target detections. Therefore, an artificially designed FPN is not necessarily the optimal structure. The combination number of feature fusion at various scales also increases with an increase in the number of network layers. Therefore, an FPN is constructed using the neural network architecture search. A search space that covers all cross-scale connections and captures multi-scale features is designed. Next, an RNN controller is obtained through reinforcement learning training to select the optimal FPN structure. The objective of the search is to find particle architectures that have the same input and output feature levels and can be applied repeatedly. The pyramid architecture can also be made manageable by modularizing the search space.

The NAS-FPN network architecture is designed based on RetinaNet and includes two parts: the Backbone Network (Basic Classification Network, MobileNet, ResNet) and Feature Pyramid Network (FPN). In NAS-FPN, a duplicate FPN module can be searched. We obtain a tradeoff between speed and accuracy by controlling the number of repetitions of this module. Then, we output the prediction results at different stages according to the different computing resources. In RetinaNet, the feature fusion strategy of FPN is adopted. NAS-FPN replaces the FPN portion of RetinaNet with the searched fusion architecture. This can find a better FPN architecture of the retinal network framework and improve the accuracy of text detection. The RetinaNet framework with NAS-FPN is depicted in Figure 4.

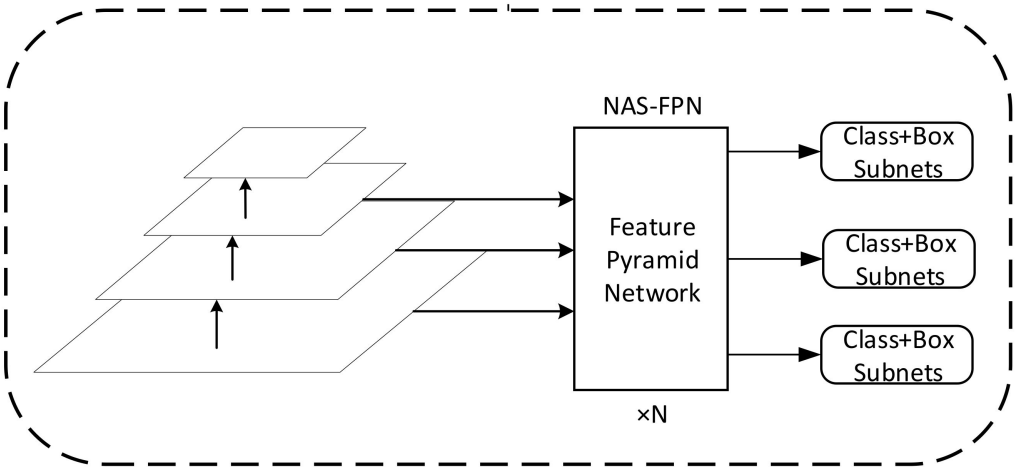



**Figure 4.** The NAS-FPN RetinaNet. N is the number of module repetitions.

A neural architecture search algorithm is used to search the FPN. The backbone network models for class and BBOX predictions follow the original design of RetinaNet. The FPN, input feature layers generated with multiple sizes, and output feature layers generated with the same size are depicted in Figure 4. The design of the RetinaNet network is adopted, using the output of the last layer in each group of networks as the input of the pyramid network. The output of the previous pyramid network is used as the input for the next pyramid network. Next, five scales are used as input features $(C1, C2, C3, C4, C5)$, and their feature steps are $(8, 16, 32, 64, 128)$, respectively. Feature C3 is obtained by the maximum pooling of C4 or C5 with strides 2 and 4. The input features are passed onto a pyramid network, which consists of a series of fused units connected across scales. The pyramid network finally outputs an enhanced multi-scale feature representation. Because both the input and output of the pyramid network are feature layers of the same size, the architecture of the FPN can be stacked $N$ times to improve the accuracy.

Multiple cross-scale connections of FPN networks can constitute a huge search space. In the search space, the FPN network is composed of many merging cells (MC), which are merged with feature representations. The fusion unit connects and fuses the feature maps from two different levels of feature layers as a feature output. The unit structure constitutes the meta-structure of the FPN network. The component feature combination of the fusion unit constitutes the search space of the algorithm. The structure of the fusion unit is depicted in Figure 5.

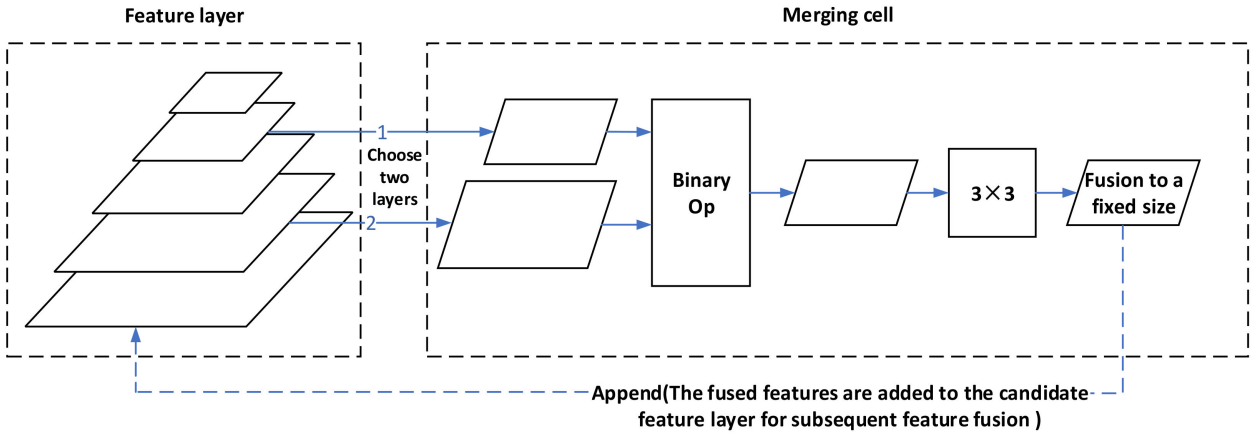



**Figure 5.** Structure of fusion unit.

The fusion unit comprises three processes: first, two feature layers are selected from the candidates; second, the dimension of the output feature is selected; and, finally, the two previously selected feature layers are combined and output to a specific scale based on the fusion method. The feature fusion methods are mainly divided into two types: sum and global pooling. Their structures are depicted in Figure 6.

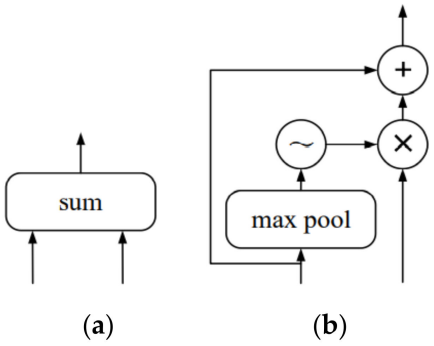



**Figure 6.** Feature fusion methods: (**a**) sum and (**b**) global pooling.

The two feature fusion methods are characterized by simplicity and effectiveness. No further trainable parameters are required. Before applying binary operations, the size of the input feature layer must be adjusted to the desired output size by upsampling the adjacent layers or by max pooling, if necessary. The merged feature layer includes a ReLu activation function, convolution operation with kernel size $3 \times 3$, and batch normalization. The NAS-FPN network selects the optimal model architecture in a given search space by training the RNN controller through reinforcement learning. The RNN controller updates the parameters using the accuracy of the submodel in the search space as a reward signal. After repeated training, the controller gradually learns how to obtain the optimal architectural model. A schematic of the learning process is depicted in Figure 7.

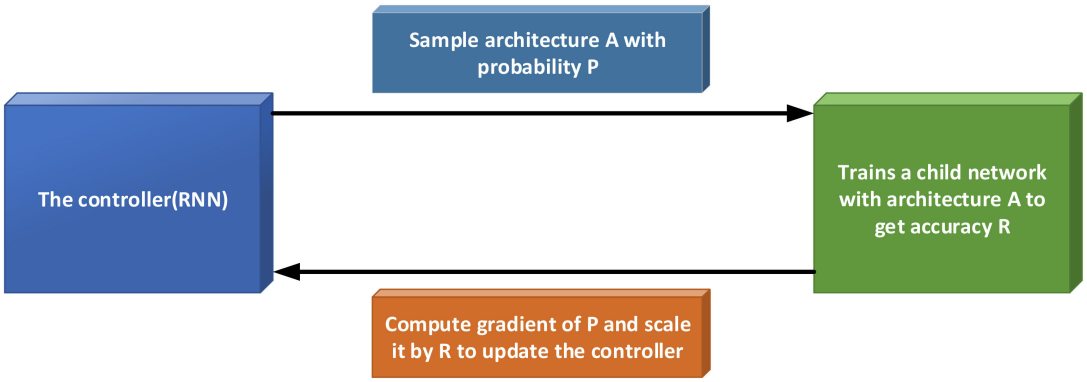



**Figure 7.** Controller reinforcement learning.

The structure and internal connections of the neural network are specified using a variable-length string. Therefore, the RNN can be used to generate variable-length network structures. The predicted network contains only convolutional layers, whose hyperparameters are generated using the RNN. These hyperparameters include the following: the height, width, and number of convolution kernels, and the height and width of the sliding stride of the convolution kernel. As depicted in Figure 8, the output predicted by each SoftMax function in the RNN is used as the next input.

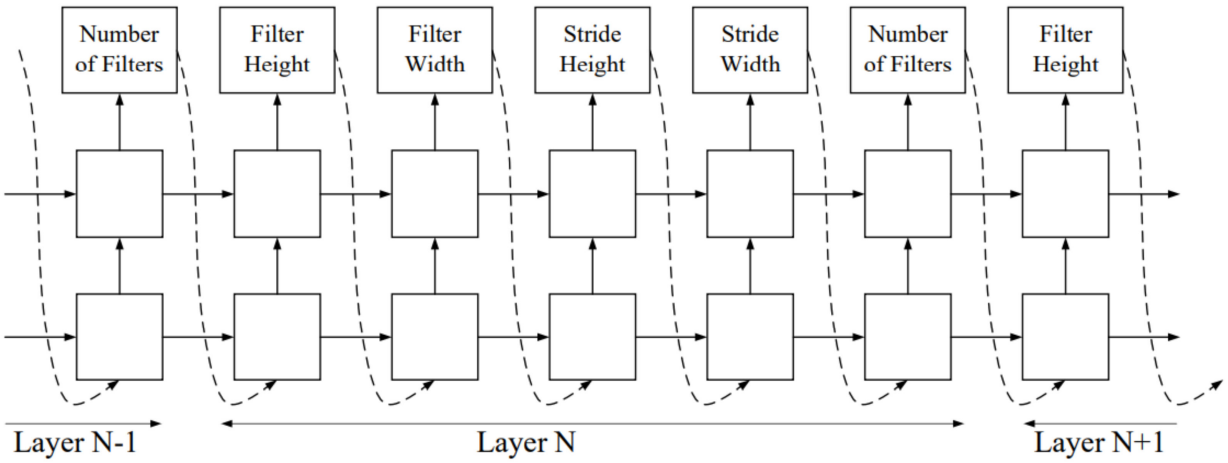



**Figure 8.** Predictive structure flowchart for RNN controller.

The controller generates a set of hyperparameters, and the accuracy of the generated model on the validation set is used as a feedback signal to optimize its expected value. The algorithm regards the actions of the controller as a function. Parameter updates are carried out via feedback signals. Then, the optimization of the feedback signal is realized. The termination condition in generating the network structure is to stop when the number of network layers reaches an optimal value.

After a parameter $\theta_c$ in the controller RNN is optimized, the resulting network structure can achieve good accuracy on the validation set.

After generating the network structure, the controller uses the training data to train until convergence. Next, the accuracy of the controller is confirmed after testing it on the validation set. The termination condition for the generated network structure is the number of network layers reaching a specified value. Parameter $\theta_c$ in the RNN controller is optimized. The generated network structure achieves improved accuracy on the verification set. The RNN controller predicts a series of outcomes, corresponding to a series of operations $a_{1:T}$ to design sub-networks. The generated network is tested on the validation set and the accuracy $R$ is obtained. The $R$ value is used as a feedback signal, and reinforcement learning is used to train the controller. To optimize the network structure, the controller is required to maximize its expected value. The formula is as follows:

$$J(\theta_c) = E_{P(a_{1:T};\theta_c)}[R] \tag{2}$$

Feedback signal $R$ is not differentiable. The policy gradient algorithm is used to iteratively update $\theta_c$ as follows:

$$\nabla\theta_c J(\theta_c) = \sum_{t=1}^{T} E_{P(a_{1t};\theta_c)}\left[\nabla\theta_c \log P\left(a_t \middle| a_{(t-1):1}; \theta_c\right) R\right] \tag{3}$$

The approximate calculation formula is as follows:

$$\nabla\theta_c J(\theta_c) = \frac{1}{m}\sum_{k=1}^{m}\sum_{t-1}^{T} \nabla\theta_c \log P\left(a_t \middle| a_{(t-1):1}; \theta_c\right) R_k \tag{4}$$

where $m$ is the number of various neural network architectures in a batch of samples during controller training, $T$ is the number of predicted hyperparameters in the controller design network structure, and $R_k$ is the tested accuracy on the validation set after the $k$th neural network is trained.

The advantage of the feature extraction network is that it designs a search space covering all possible cross-scale connections, which can be used to generate multi-scale feature representations. Particle architectures that have the same input and output feature levels and can be applied repeatedly are found during the search process. The modular search space also makes the search pyramid architecture manageable. Furthermore, feature extraction network models employing automatic architecture search algorithms can obtain feature pyramid representations at the output of any given pyramid network. The network model does not need to complete the forward pass of all pyramid networks, providing a solution that can dynamically allocate computational resources to generate detections. Therefore, an optimal FPN structure can be obtained to extract the feature of the text region.

2.2.2. Generalized Intersection over Union Algorithm

Currently, the optimization of a BBOX in the text detection method is performed mostly by reducing the regression loss of the BBOX. In this study, we trained on regression tasks with IoU as a direct metric. In the anchor-node-based target detection method, the IoU can be used not only to determine the positive and negative samples but also to judge the accuracy of the prediction frame. In addition, the IoU is insensitive to scale. Assuming that A and B denote the predicted box and labeled text area, respectively, the IoU and loss function are given as follows:

$$\mathrm{IOU} = \frac{|\mathrm{A} \cap \mathrm{B}|}{|\mathrm{A} \cup \mathrm{B}|} \tag{5}$$

$$\mathrm{Loss}_{\mathrm{giou}} = 1 - \mathrm{IOU} \tag{6}$$

There are two problems with using the IoU directly as the loss function. First, when the two boxes do not intersect, IOU $= 0$ can be obtained by definition. In this case, the contact ratio of the two boxes cannot be reflected. Second, when Loss $= 0$, model training cannot be performed because there is no gradient backhaul. As depicted in Figure 9, when the IoU is the same, the detection results of the prediction boxes may also be different. The text detection result on the far right is the worst.

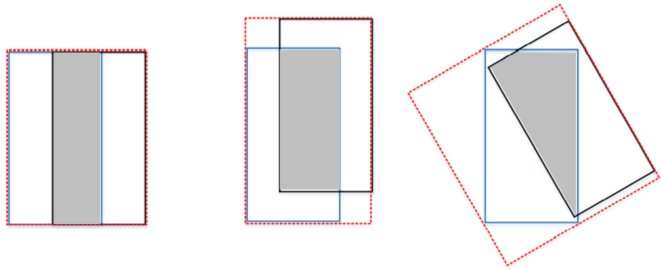


**Figure 9.** Text detection results with the same IoU value.

Because the use of IoU has many drawbacks, we propose to use another metric, GIoU, which is given as follows:

$$\mathrm{GIoU}(\mathrm{A}, \mathrm{B}) = \mathrm{IoU}(\mathrm{A}, \mathrm{B}) - \frac{|\mathrm{C}| - |\mathrm{AUB}|}{|\mathrm{C}|} \tag{7}$$

As shown in Figure 10, A is the prediction box, B is the real box, and S is the set of all boxes. Whether A and B intersect or not, C is the smallest box containing A and B (the smallest convex closed box containing A and B), and C also belongs to the S set.

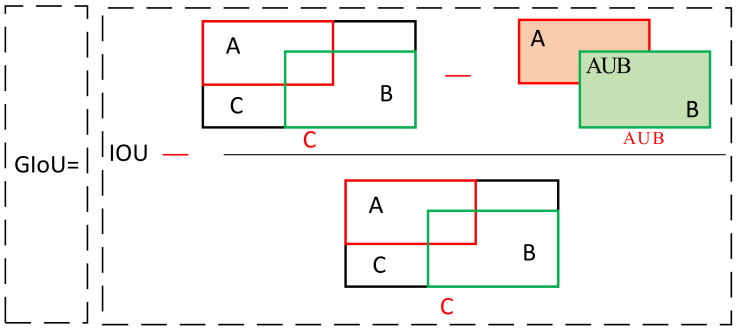



**Figure 10.** Structure diagram of GIoU.

We calculate IoU (the ratio of the intersection of A and B), and then calculate the area ratio of the C (no A and B) to C. Finally, the GIoU is obtained by subtracting the specific gravity value from the IoU.

The coordinates of prediction box $B^p$ and marker box $B^G$ are as follows:

$$B^P = \left(x_1^P,\ y_1^P, x_2^P, y_2^P\right), B^G = \left(x_1^G, y_1^G, x_2^G, y_2^G\right) \tag{8}$$

For prediction box $B^p$, ensure that $x_2^P > x_1^P$ and $y_2^P > y_1^P$:

$$\begin{array}{l} \hat{x}_1^p = \min\left(x_1^P, x_2^P\right) \\ \hat{x}_2^p = \max\left(x_1^P, x_2^P\right) \\ \hat{y}_1^p = \min\left(y_1^P, y_2^P\right) \\ \hat{y}_2^p = \max\left(y_1^P, y_2^P\right) \end{array} \tag{9}$$

The areas of predicted box $B^p$ and labeled box $B^G$ are calculated as follows:

$$\begin{array}{l} A^P = \left(\hat{x}_2^p - \hat{x}_1^p\right) \times \left(\hat{y}_2^p - \hat{y}_1^p\right) \\ A^G = \left(x_2^p - x_1^p\right) \times \left(y_2^p - y_1^p\right) \end{array} \tag{10}$$

The coordinates and area of area $I$ where $B^p$ and $B^G$ intersect are calculated as follows:

$$\begin{array}{l} x_1^I = \min\left(x_1^I, x_2^I\right) \\ x_2^I = \max\left(x_1^I, x_2^I\right) \\ y_1^I = \min\left(y_1^I, y_2^I\right) \\ y_2^I = \max\left(y_1^I, y_2^I\right) \end{array} \tag{11}$$

$$I = \begin{cases} \left(x_2^I - x_1^I\right) \times \left(y_2^I - y_1^I\right) & \text{if} \quad x_2^I > x_1^I, y_2^I > y_1^I \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

The same method is used to obtain the coordinates and area $A^C$ of the smallest BBOX $C$. The area of the text area is defined as $U$, and the value of $U$ is calculated as follows:

$$U = A^p + A^G - I \tag{13}$$

GIoU is calculated as follows:

$$\text{GIOU} = \frac{I}{U} - \frac{A^C - U}{A^C} \tag{14}$$

When GIoU is used as the loss function, $L_{\text{GIOU}} = 1 - \text{GIOU}$, which can meet the basic requirements of the loss function. In addition, GIoU is also independent of size. It is the lower bound of IoU. When the two boxes overlap infinitely, IOU = GIOU and the value is between (0, 1). However, the value range of GIoU is $(-1, 1)$. When the positions of the

two boxes overlap completely, the maximum value is 1; when the two boxes do not overlap fully, the minimum value is −1. Therefore, GIoU is a very good distance metric. However, unlike IoU, GIoU calculates not only the overlapping part but also the non-overlapping area. In addition, GIoU calculates the overlapping area in the same manner as does the IoU in text detection. To calculate the minimum closure area, only the maximum and minimum coordinates are required.

### *2.3. Class Imbalance*

Class imbalance refers to the significant difference in the numbers of training examples of various classes in the target classification [43]. Owing to the limited number of samples with BBOX as the target category in an image, the results of the statistical analysis of the ICDAR2013 [44] dataset are as depicted in Figure 11. The text areas in most images occupy only 30% of the entire image. Most of the widths are also concentrated between 0 and 0.3. The height ratio is concentrated between 0 and 0.15. Therefore, the proportion of text areas in natural scene images is generally small. Furthermore, the shape of the area is narrow and long.

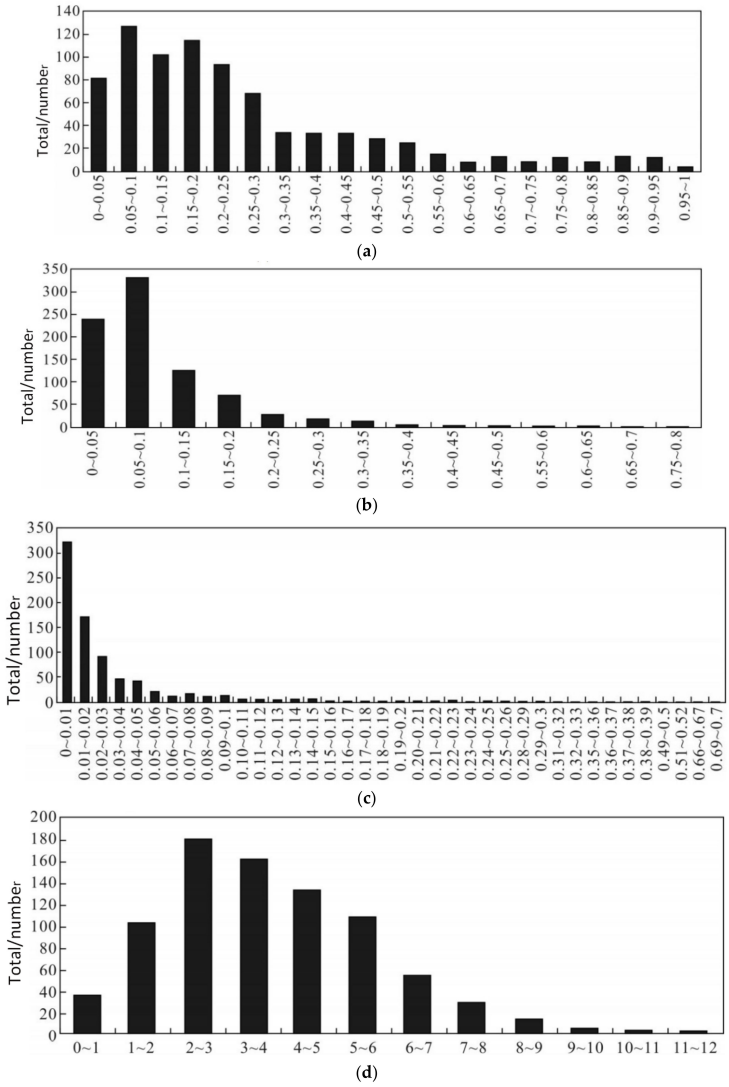



**Figure 11.** Characteristic analysis of text area in ICDAR2013 database. (**a**) Width of text area/width of full image, (**b**) height of text area/width of full image, (**c**) text regional area/full image area, and (**d**) width of text area/height of text area.

As depicted in Figure 12, the BBOX required to calculate the loss value is divided into two types: positive and negative. When the GIoU between the BBOX and ground truth (GT) is greater than a threshold value, the BBOX is a positive BBOX. When GIoU is less than the threshold value, the BBOX is a negative BBOX.

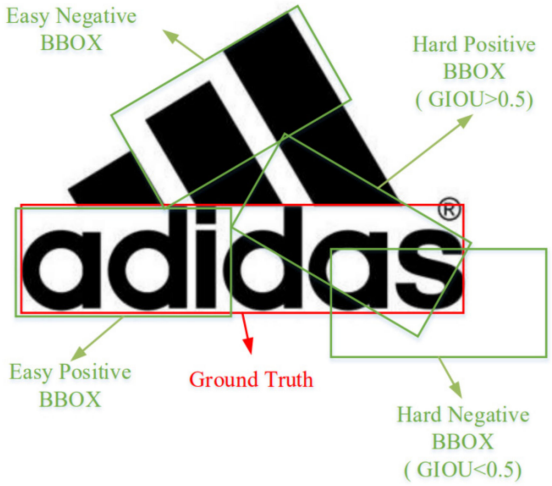



**Figure 12.** Schematic of four samples.

When an original image is input, the proportion of the target is only a small part of the whole image. Therefore, the two types of BBOXs are mainly negative, and most of the negative BBOXs are not in the transition area between the foreground and background. This clearly classified negative BBOX is called easy negative and is common. This leads to two problems:

1. An extremely negative BBOX will cause its loss value to be significantly large. The loss value of a positive BBOX is overwhelmed, which is not conducive to the convergence of the target.
2. When the parameter changes in the training process are not evident, the model cannot be effectively trained, and the problem of gradient disappearance may occur. However, when the easy negative sample is trained, the corresponding target score is small. That is, the loss value of a single BBOX sample is small. The parameter changes during model training backpropagation are also significantly small. Small parameter changes are not conducive to model training. Therefore, for text detection, it is extremely necessary to find BBOX samples with larger loss values and a greater impact on parameter convergence—namely, a hard BBOX.

The detection and analysis of the EAST algorithm indicate that class imbalance is another reason for its poor performance. Therefore, it is necessary to reduce the proportion of simple sample loss values. Furthermore, the sample loss values with confidence greater than 0.5 must be suppressed.

The cross-entropy (CE) loss for binary classification is calculated as follows:

$$CE(p,y) = \begin{cases} -\log(p) & y = 1 \\ -\log(1-p) & \text{otherwise} \end{cases} \tag{15}$$

where $y = 1$ represents a positive sample and the $p$ value is the model's estimated probability for the class with the $y = 1$ label. Parameter $p_t$ is defined as follows:

$$p_t = \begin{cases} p & y = 1 \\ 1-p & \text{otherwise} \end{cases} \tag{16}$$

$$CE(p,\mathrm{y}) = CE(p_t) = -\log(p_t) \tag{17}$$

A parameter is added to control and balance the proportion of positive and negative sample loss values:

$$CE(p_t) = -\alpha_t \log(p_t) \tag{18}$$

Although the proportion of positive and negative samples is balanced, easy and difficult examples are not distinguished. A modulation factor $(1 - p_t)^\gamma$ is added to the CE loss function, where $\gamma \geq 0$. The FL loss function is defined as

$$FL(P_t) = -(1 - P_t)^\gamma \log(p_t) \tag{19}$$

Figure 13 is the FL diagram at the $\gamma = 0,\ 1,\ 2,\ 5$ value, whereas the CE loss is represented by curve 1.

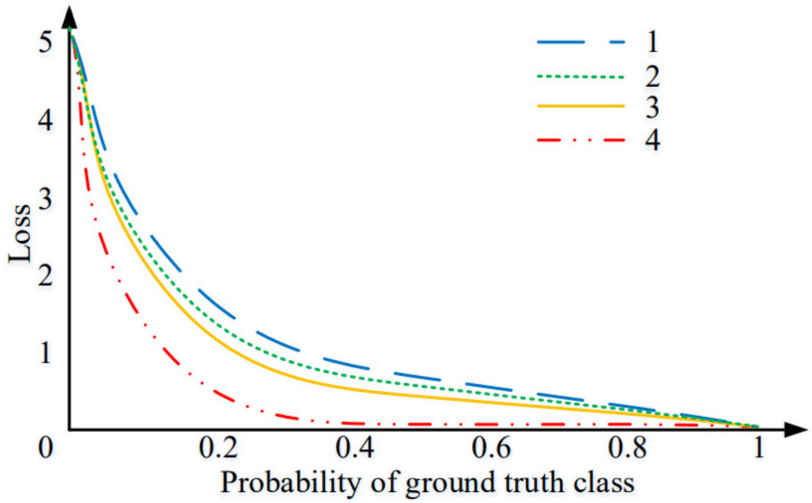



**Figure 13.** Loss function change curve.

As depicted in Figure 13, when the example is misclassified and $p_t$ is small, modulation factor $(1 - p_t)^\gamma$ is close to 1 and the loss is not affected. When $p_t \to 1$, the factor becomes 0 and the loss on well-classified examples is weighted. Furthermore, the focus parameter z smoothly adjusts the weights of simple examples. When $\gamma = 0$, FL is equal to CE. As the parameter increases, the effect of the modulation factor also increases. For the third curve $\gamma = 2$, FL works the best. The modulation factor reduces the loss contribution in the simple example. Furthermore, the range of the sample reception low loss is extended. Finally, the b-balanced variant of FL is used.

The $\alpha$ of *FL* is used to balance the variant:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \tag{20}$$

This approach improves the accuracy when compared with the non-$\alpha$-balanced form. AP is the average precision. The formula is as follows:

$$\sum_{k=1}^{N} P(k)\Delta r(k) \tag{21}$$

where $N$ represents the number of all pictures in the test set; $P(k)$ represents the precision value when $k$ images can be recognized. Delta $r(k)$ represents the change in the recall value when the number of recognized images changes from $k - 1$ to $k$.

### *2.4. Experimental Dataset*

The datasets that we used are ICDAR 2013 and ICDAR 2015 [45]. Resolutions range from 720 × 480 to 1280 × 960. They include real-world images (road signs, billboards, posters, etc.). ICDAR 2013 contains 229 training images and 233 test images of focused scene text. Text is in English and aligned horizontally. Annotations are axis-aligned bounding boxes that divide a total of 1015 cropped word images. ICDAR 2015 contains 1000 training images and 500 test images. Annotations are word-level quadrilateral tilted text. The details of the datasets are listed in Table 2.

**Table 2.** Experimental datasets.

| Dataset | Size | Number of Images (Train/Test) | Amount of Text |
|---|---|---|---|
| ICDAR 2013 | 250 M | 462 (229/233) | 1943 |
| ICDAR 2015 | 131.8 M | 1500 (1000/500) | 17,548 |

## 3. Results

### *3.1. Impact of Pyramid Network on Text Detection*

The text detection model can be controlled by adjusting the dimensions of the backbone model and pyramid network. The model was trained and evaluated on the ICDAR2015 dataset. The training parameters were set as listed in Table 3.

**Table 3.** Training parameter settings.

| Type | Setting |
|---|---|
| Batch size | 16 |
| learning rate | $10^{-3}$ |
| Focal loss | $a = 0.25, \gamma = 2$ |
| Learning decay rate | 0.9/10,000 |
| Iterations | 100,000 |

Pyramid networks offer the advantage that they can be extended to larger architectures by stacking multiple repeated architectures. We first changed the number of FPNs to test their impact on text detection. The experimental results are depicted in Figure 14a, where the numbers on the lines indicate the number of pyramid networks. Stacking FPN architectures does not always improve the performance of the model; however, stacking NAS-FPN can significantly improve the accuracy. This result indicates that the proposed search algorithm can find a scalable pyramid network architecture suitable for text detection.

The backbone architecture has a significant impact on the pyramid network. We performed comparative experiments with the ResNet-50, ResNet-101, and MobilenetV2 backbone networks. The results of these experiments are depicted in Figure 14b. When the number of pyramid networks increases, the performance of NAS-FPN on all these backbone architectures increases. The results indicate that the text detection model can work well with various architectures. However, the text detection is better on the ResNet network.

### *3.2. Impact of Modules on Text Detection*

The NEAST text detection model used in this study is an improved text detection algorithm based on the EAST algorithm. It has an improved FPN, addresses the problem of class imbalance, and corrects the IoU of the output module. To evaluate the impact of each module on the entire algorithm model, the modules of the model were disabled in turn on the ICDAR 2015 dataset to compare the recall, precision, and F-Measure values. The comparative data are listed in Table 4, where *Tt* represents the time spent in training the model and *Dt* is the time spent in detecting text.

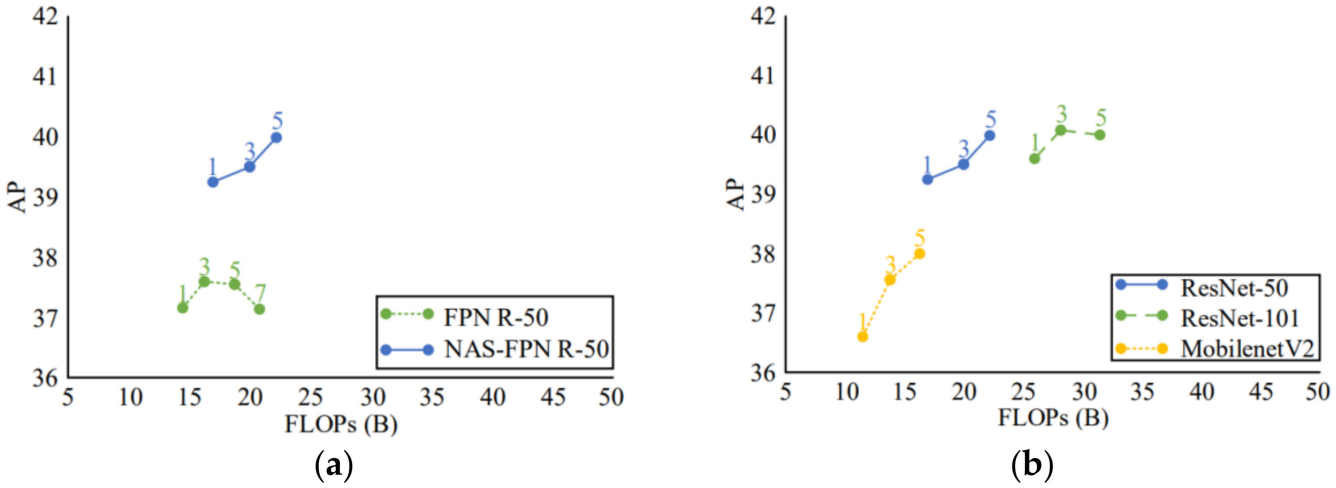



**Figure 14.** Impact of NAS-FPN on text detection. Impact of (**a**) search algorithm and (**b**) backbone network on model performance.

**Table 4.** Experimental results for various modules.

| Method | Tt/h | Dt/ms | R | P | F |
|---|---|---|---|---|---|
| EAST | 19.7 | 150.9 | 0.73 | 0.84 | 0.78 |
| FL+GIOU | 33.4 | 198.0 | 0.79 | 0.85 | 0.82 |
| NAS+GIOU | 139.3 | 307.8 | 0.76 | 0.87 | 0.81 |
| NAS+FL | 144.9 | 323.5 | 0.82 | 0.85 | 0.83 |
| Proposed | 145.6 | 337.4 | 0.84 | 0.89 | 0.87 |

The proposed method takes the longest times for model training and text detection because the temporal neural network architecture search algorithm and class equalization process both generate many parameters and increase the complexity of the model. However, the proposed method is the best in terms of the recall, precision, and comprehensiveness metrics. The FL function addresses the problem of class imbalance and mitigates the problem of low recall rate during object detection. The GIoU algorithm can further improve the accuracy of text detection. Although the GIoU algorithm generates fewer additional parameters, the accuracy of the text detection model is significantly improved. Experimental results confirmed that the proposed neural network architecture search algorithm achieved significantly improved detection results.

## 4. Discussion

A comparison of the text detection performance between the proposed model and some commonly used text detection algorithms on the ICDAR 2013 and ICDAR 2015 datasets was conducted in this work. By implementing the model on different datasets, the applicability and generalization parameters of the model were evaluated.

We discuss fine-scale detection strategies for FASText [46] and Faster R-CNN [47]. As shown in Table 5, it is difficult for a single RPN to perform accurate text localization due to the large amount of error detection (low accuracy). Improvements were made to the RPN algorithm by using the Fast R-CNN detection model, which significantly improves the positioning accuracy, with an F-measure of 0.73. Faster R-CNN also improves the recall of the original RPN. This may benefit from the joint bounding box regression mechanism of Fast R-CNN, which improves the accuracy of bounding box prediction. Although FASText can roughly locate the text line or the main part of a word, it cannot capture the most precise location compared to the ICDAR 2013 standard. Obviously, the proposed NEAST significantly improves Faster R-CNN and FASText in terms of accuracy and recall, which indicates that NEAST predicts a fine-scale text proposal of a sequence rather than a whole

line of text. Nonetheless, our detection algorithm is more accurate and reliable. As can be seen from the data in Table 5, our proposed detection algorithm achieves the optimal recall, accuracy, and F-measure values on the ICDAR2013 and IDDAR2015 datasets.

**Table 5.** Comparison of text detection results of various methods on the ICDAR 2013 dataset.

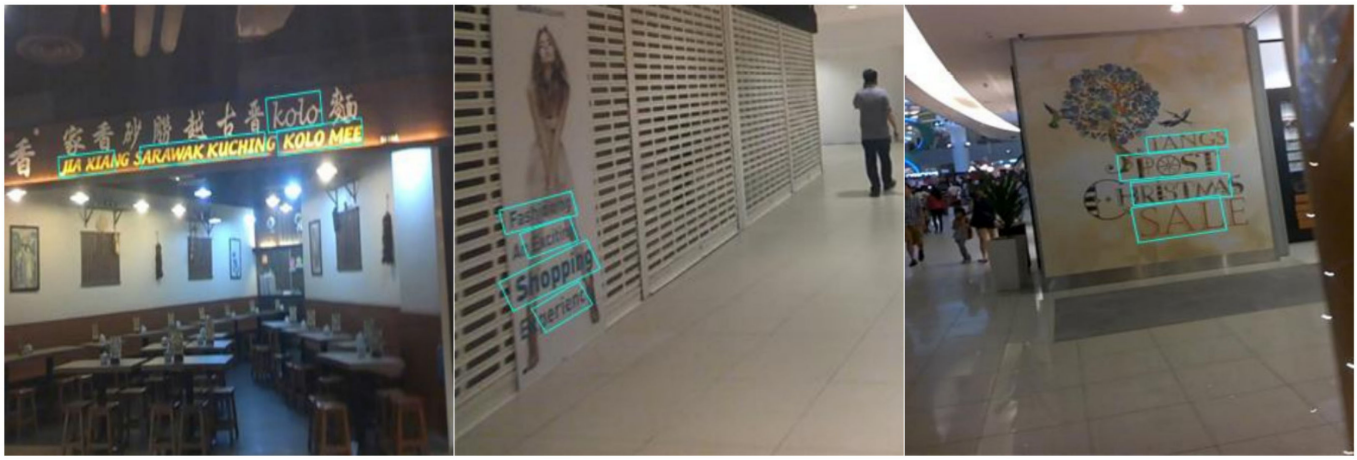
| Method | R | P | F |
|---|---|---|---|
| Ren [46] | 0.67 | 0.81 | 0.73 |
| Faster RCNN [47] | 0.68 | 0.72 | 0.73 |
| SSD [48] | 0.60 | 0.80 | 0.68 |
| Diaz-Escobar [49] | 0.74 | 0.83 | 0.78 |
| Proposed | 0.81 | 0.86 | 0.83 |

The complexity and quantity of natural scenes and images in different datasets are different. The results based on different datasets show that the NEAST text detection algorithm can achieve better detection results and is more robust in complex backgrounds. We discussed the impact of repeated connections on NEAST. Compared with the data listed in Table 6, it is clear that the proposed text detection method performs better on the ICDAR 2015 dataset. Therefore, the conclusion is that the larger the training dataset, the better the text detection model. Contextual information helps to reduce error detection, such as text outliers, which is important for recovering highly ambiguous text—for example, very small text. It is one of the main strengths of our NEAST, which can dramatically increase the F-measure of EAST from 0.78 to 0.87 through our repeated connections.

**Table 6.** Comparison of text detection results of various methods on the ICDAR 2015 dataset.

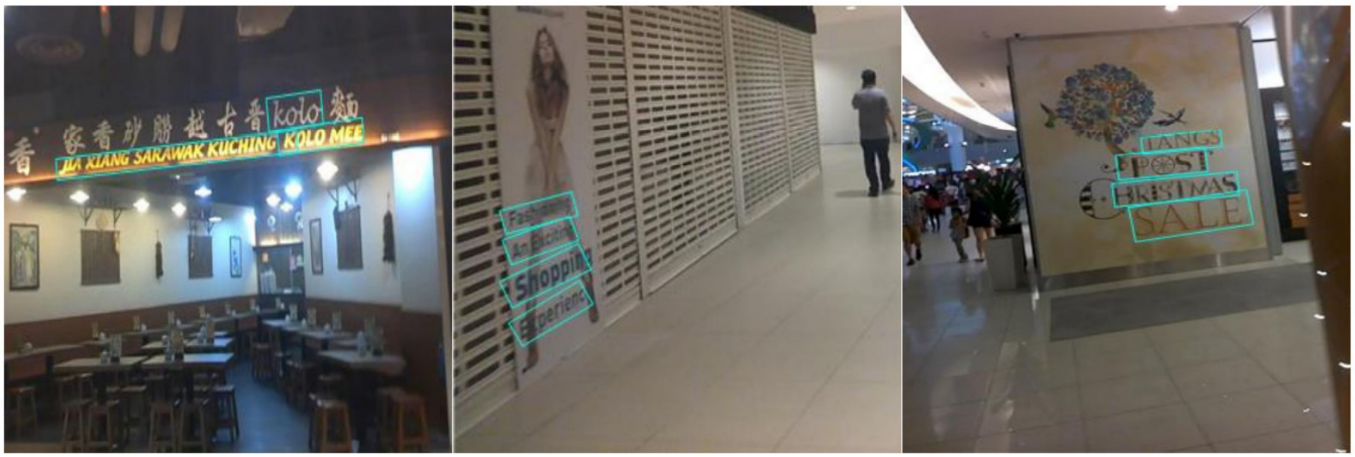
| Method | R | P | F |
|---|---|---|---|
| Xue [50] | 0.78 | 0.86 | 0.82 |
| EAST [39] | 0.73 | 0.84 | 0.78 |
| SSTD [51] | 0.74 | 0.80 | 0.77 |
| Jiang [52] | 0.83 | 0.87 | 0.85 |
| R2CNN [27] | 0.80 | 0.85 | 0.83 |
| PixelLink [53] | 0.82 | 0.85 | 0.84 |
| Proposed | 0.84 | 0.89 | 0.87 |

From the ICDAR2015 dataset, we selected three natural scene images with high background complexity, oblique text, and multiple scales. Experiments were performed on the models with and without the improvement of the EAST algorithm to compare their text detection performance. Figures 15 and 16 depict the text detection performance obtained using the EAST algorithm without and with improvement, respectively. It is evident that the proposed improved text detection model can accurately detect the text regions in horizontal and oblique directions, and the detection effect on long texts is also satisfactory. Through targeted training, the proposed detection method can more accurately exclude non-text regions, thereby improving the accuracy of the final detection results. It is also evident that the performance of the proposed algorithm for oblique text detection in any direction is significantly improved when compared with that of the EAST algorithm.




(**a**) (**b**) (**c**)

**Figure 15.** Text detection results of EAST algorithm: (**a**) test image 1, (**b**) test image 2, and (**c**) test image 3.




(**a**) (**b**) (**c**)

**Figure 16.** Text detection results of proposed improved EAST algorithm: (**a**) test image 1, (**b**) test image 2, and (**c**) test image 3.

The performance of the proposed NEAST text detection algorithm was evaluated on complex natural scenes with oblique multi-scale text. The results are depicted in Figure 17a–c. The algorithm accurately identified multi-scale text at various shooting angles. As depicted in Figure 17d–f, the algorithm can also detect text in scenes with dynamic light conditions. These results confirm the high detection strength and accuracy of the proposed algorithm against complex scenes.


**Figure 17.** Oblique text detection results in natural scenes.

## 5. Conclusions

In this paper, an improved NEAST algorithm is proposed for skewed text detection in complex scenes and images with multi-scale text. We propose the following innovations. Firstly, we use the Learning Scalable Feature Pyramid Architecture for Object Detection (NAS-FPN) to integrate features of multiple layers and construct an FPN by means of a neural architecture search. By designing a search space that can cover all cross-scale connections and acquire multi-scale features, an RNN controller is acquired through reinforcement learning training to select the optimal FPN structure. Secondly, the GIoU algorithm is used to replace the IoU algorithm, so as to improve the regression of text BBOX. Finally, the focal loss function is used to resolve the class imbalance problem. The core idea of our proposed algorithm is to obtain a network structure in the FPN search space by an RNN controller, set it as a subnetwork, and then use this network structure to train on the dataset. The final accuracy is obtained by testing on the validation set. This accuracy rate is applied on the controller, and the controller continues to optimize to obtain another network structure. This practice is repeated until the best feature extraction network structure is obtained, and, finally, the detection result of skewed text is realized. In future research, we will continue

to improve the NEAST text recognition algorithm by targeting the diversity of languages, including Chinese and numbers. The unsupervised ultra-lightweight backbone network will be used to mine the deep semantic information of complex natural scenes and images.

**Author Contributions:** Conceptualization, Y.P. and G.Z.; methodology and guidance of the project, W.S., Y.P. and G.Z.; validation, formal analysis, and data analysis, W.S. and G.Z.; writing, W.S., H.W., Y.L., J.L. (Jiasai Luo), T.L., J.L. (Jinzhao Lin), Y.P. and G.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Doctoral Innovative Talents Project of Chongqing University of Posts and Telecommunications (BYJS202107), the Natural Science Foundation of Chongqing (grant number cstc2021jcyj-bsh0218); the Science and Technology Bureau of Chongqing (D63012021013); the National Natural Science Foundation of China (Grant No. U21A20447 and 61971079); the Basic Research and Frontier Exploration Project of Chongqing (Grant No. cstc2019 jcyjmsxmX0666); the Chongqing Technological Innovation and Application Development Project (cstc2021jscx-gksbx0051); the Innovative Group Project of the National Natural Science Foundation of Chongqing (Grant No. cstc2020jcyj-cxttX0002); the Regional Creative Cooperation Program of Sichuan (2020YFQ0025); and the Science and Technology Research Program of Chongqing Municipal Education Commission (KJZD-k202000604).

**Data Availability Statement:** Not applicable.

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

## References

1. 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 (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7098–7107.
2. Wu, F.; Zhu, C.; Xu, J.; Bhatt, M.W.; Sharma, A. Research on image text recognition based on canny edge detection algorithm and k-means algorithm. *Int. J. Syst. Assur. Eng.* **2022**, *13*, 72–80. [CrossRef]
3. Kisacanin, B.; Pavlovic, V.; Huang, T.S. *Real-Time Vision for Human-Computer Interaction*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2015.
4. Barber, D.B.; Redding, J.D.; McLain, T.W.; Beard, R.W.; Taylor, C.N. Vision-based target geo-location using a fixed-wing miniature air vehicle. *J. Intell. Robot. Syst.* **2006**, *47*, 361–382. [CrossRef]
5. Haritaoglu, I. Scene text extraction and translation for handheld devices. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA, 8–14 December 2001; p. II.
6. DeSouza, G.N.; Kak, A.C. Vision for mobile robot navigation: A survey. *IEEE Tran. Pattern Anal.* **2002**, *24*, 237–267. [CrossRef]
7. Ham, Y.K.; Kang, M.S.; Chung, H.K.; Park, R.H.; Park, G.T. Recognition of raised characters for automatic classification of rubber tires. *Opt. Eng.* **1995**, *34*, 102–109. [CrossRef]
8. Neumann, L.; Matas, J. Real-time lexicon-free scene text localization and recognition. *IEEE Tran. Pattern Anal.* **2015**, *38*, 1872–1885. [CrossRef]
9. Louloudis, G.; Gatos, B.; Pratikakis, I.; Halatsis, C. Text line detection in handwritten documents. *Pattern Recogn.* **2008**, *41*, 3758–3772. [CrossRef]
10. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
11. Jinqiang, W.; Basnet, P.; Mahtab, S. Review of machine learning and deep learning application in mine microseismic event classification. *Min. Miner. Deposits* **2021**, *15*, 19–26. [CrossRef]
12. Peng, P.; He, Z.; Wang, L.; Jiang, Y. Automatic classification of microseismic records in underground mining: A deep learning approach. *IEEE Access* **2020**, *8*, 17863–17876. [CrossRef]
13. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agr.* **2018**, *147*, 70–90. [CrossRef]
14. Jospin, L.V.; Laga, H.; Boussaid, F.; Buntine, W.; Bennamoun, M. Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Comput. Intell. Mag.* **2022**, *17*, 29–48. [CrossRef]
15. Su, X.; Xue, S.; Liu, F.; Wu, J.; Yang, J.; Zhou, C.; Hu, W.; Paris, C.; Nepal, S.; Jin, D.; et al. A Comprehensive Survey on Community Detection with Deep Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**. *Early Access*. [CrossRef]
16. Li, T.; Wang, Y.; Hong, R.; Wang, M.; Wu, X. pDisVPL: Probabilistic discriminative visual Part Learning for image classification. *IEEE MultiMedia* **2018**, *25*, 34–45. [CrossRef]
17. Li, T.; Cheng, B.; Ni, B.; Liu, G.; Yan, S. Multitask low-rank affinity graph for image segmentation and image annotation. *ACM T. Intel. Syst. Tec.* **2016**, *7*, 1–18. [CrossRef]
18. Li, T.; Ni, B.; Xu, M.; Wang, M.; Gao, Q.; Yan, S. Data-driven affective filtering for images and videos. *IEEE T. Cybernetics* **2015**, *45*, 2336–2349. [CrossRef]

19. Wang, W.; Xie, E.; Li, X.; Hou, W.; Lu, T.; Yu, G.; Shao, S. Shape Robust Text Detection with Progressive Scale Expansion Network. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9328–9337.
20. Wang, T.; Wu, D.J.; Coates, A.; Ng, A.Y. End-to-end text recognition with convolutional neural networks. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 3304–3308.
21. Zang, D.; Zhang, J.; Zhang, D.; Bao, M.; Cheng, J.; Tang, K. Traffic sign detection based on cascaded convolutional neural networks. In Proceedings of the 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Shanghai, China, 30 May–1 June 2016; pp. 201–206.
22. Zhang, Z.; Zhang, C.; Shen, W.; Yao, C.; Liu, W.; Bai, X. Multi-oriented Text Detection with Fully Convolutional Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4159–4167.
23. Liu, X.; Liang, D.; Yan, S.; Chen, D.; Qiao, Y.; Yan, J. FOTS: Fast Oriented Text Spotting with a Unified Network. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5676–5685.
24. He, W.; Zhang, X.; Yin, F.; Liu, C. Deep Direct Regression for Multi-oriented Scene Text Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 745–753.
25. Qin, S.; Ren, P.; Kim, S.; Manduchi, R. Robust and Accurate Text Stroke Segmentation. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 242–250.
26. Feng, W.; He, W.H.; Yin, F.; Liu, C.L. Scene Text Detection with Recurrent Instance Segmentation. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 2227–2232.
27. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Fu, P.; Luo, Z. $R^2$ CNN: Rotational Region CNN for Arbitrarily-Oriented Scene Text Detection. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 3610–3615.
28. Liao, M.; Zhu, Z.; Shi, B.; Xia, G.; Bai, X. Rotation-Sensitive Regression for Oriented Scene Text Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5909–5918.
29. Liu, X.; Zhou, G.; Zhang, R.; Wei, X. An Accurate Segmentation-Based Scene Text Detector with Context Attention and Repulsive Text Border. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 2344–2352.
30. Long, S.; Ruan, J.; Zhang, W.; He, X.; Wu, W.; Yao, C. TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes. In Proceedings of the Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 19–35.
31. Lyu, P.; Yao, C.; Wu, W.; Yan, S.; Bai, X. Multi-oriented Scene Text Detection via Corner Localization and Region Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7553–7563.
32. Nayef, N.; Yin, F.; Bizid, I.; Choi, H.; Feng, Y.; Karatzas, D.; Luo, Z.; Pal, U.; Rigaud, C.; Chazalon, J.; et al. ICDAR2017 Robust Reading Challenge on Multi-Lingual Scene Text Detection and Script Identification-RRC-MLT. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; pp. 1454–1459.
33. Li, J.; Lin, Y.; Liu, R.; Ho, C.M.; Shi, H. RSCA: Real-time Segmentation-based Context-Aware Scene Text Detection. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 19–25 June 2021; pp. 2349–2358.
34. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
35. Huang, W.; Qiao, Y.; Tang, X. Robust scene text detection with convolution neural network induced mser trees. In Proceedings of the European Conference on Computer Vision (ECCV), Zürich, Switzerland, 6–7 and 12 September 2014; pp. 497–511.
36. Jaderberg, M.; Vedaldi, A.; Zisserman, A. Deep features for text spotting. In Proceedings of the European Conference on Computer Vision (ECCV), Zürich, Switzerland, 6–7 and 12 September 2014; pp. 512–528.
37. Shi, B.; Bai, X.; Belongie, S. Detecting Oriented Text in Natural Images by Linking Segments. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3482–3490.
38. Tian, Z.; Huang, W.; He, T.; He, P.; Qiao, Y. Detecting text in natural image with connectionist text proposal network. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 56–72.
39. Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; Liang, J. East: An efficient and accurate scene text detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5551–5560.
40. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
41. Hou, F.; Lei, W.; Li, S.; Xi, J.; Xu, M.; Luo, J. Improved Mask R-CNN with distance guided intersection over union for GPR signature detection and segmentation. *Automat. Constr.* **2021**, *121*, 103414. [CrossRef]

42. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
43. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE T. Multimed.* **2018**, *20*, 3111–3122. [CrossRef]
44. Stamatopoulos, N.; Gatos, B.; Louloudis, G.; Pal, U.; Alaei, A. ICDAR 2013 Handwriting Segmentation Contest. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 1402–1406.
45. 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 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; pp. 1156–1160.
46. Buta, M.; Neumann, L.; Matas, J. FASText: Efficient Unconstrained Scene Text Detector. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1206–1214.
47. Kumar, A.; Zhang, Z.J.; Lyu, H. Object detection in real time based on improved single shot multi-box detector algorithm. *EURASIP J. Wirel. Comm.* **2020**, *2020*, 204. [CrossRef]
48. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *ITPAM* **2017**, *39*, 1137–1149. [CrossRef]
49. Diaz-Escobar, J.; Kober, V. Natural Scene Text Detection and Segmentation Using Phase-Based Regions and Character Retrieval. *Math. Probl. Eng.* **2020**, *2020*, 7067251. [CrossRef]
50. Xue, C.; Lu, S.; Zhang, W. MSR: Multi-scale shape regression for scene text detection. *arXiv* **2019**, arXiv:1901.02596.
51. He, P.; Huang, W.; He, T.; Zhu, Q.; Qiao, Y.; Li, X. Single Shot Text Detector with Regional Attention. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3066–3074.
52. Jiang, X.; Xu, S.; Zhang, S.; Cao, S. Arbitrary-Shaped Text Detection with Adaptive Text Region Representation. *IEEE Access* **2020**, *8*, 102106–102118. [CrossRef]
53. Deng, D.; Liu, H.; Li, X.; Cai, D. Pixellink: Detecting scene text via instance segmentation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 7–8 February 2018; Volume 32.