information

Article

CSFF-Net: Scene Text Detection Based on Cross-Scale Feature Fusion

Yuan Li, Mayire Ibrayim * and Askar Hamdulla

College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; liyuan96@stu.xju.edu.cn (Y.L.); askar@xju.edu.cn (A.H.)
* Correspondence: mayire401@xju.edu.cn; Tel.: +86-133-1988-9043

Abstract: In the last years, methods for detecting text in real scenes have made significant progress with an increase in neural networks. However, due to the limitation of the receptive field of the central nervous system and the simple representation of text by using rectangular bounding boxes, the previous methods may be insufficient for working with more challenging instances of text. To solve this problem, this paper proposes a scene text detection network based on cross-scale feature fusion (CSFF-Net). The framework is based on the lightweight backbone network Resnet, and the feature learning is enhanced by embedding the depth weighted convolution module (DWCM) while retaining the original feature information extracted by CNN. At the same time, the 3D-Attention module is also introduced to merge the context information of adjacent areas, so as to refine the features in each spatial size. In addition, because the Feature Pyramid Network (FPN) cannot completely solve the interdependence problem by simple element-wise addition to process cross-layer information flow, this paper introduces a Cross-Level Feature Fusion Module (CLFFM) based on FPN, which is called Cross-Level Feature Pyramid Network (Cross-Level FPN). The proposed CLFFM can better handle cross-layer information flow and output detailed feature information, thus improving the accuracy of text region detection. Compared to the original network framework, the framework provides a more advanced performance in detecting text images of complex scenes, and extensive experiments on three challenging datasets validate the realizability of our approach.

Keywords: feature extraction; attention mechanism; pyramid network; deep learning; text detection

check for updates

Citation: Li, Y.; Ibrayim, M.; Hamdulla, A. CSFF-Net: Scene Text Detection Based on Cross-Scale Feature Fusion. *Information* **2021**, *12*, 524. https://doi.org/10.3390/info12120524

Academic Editor: Ralf Krestel

Received: 15 November 2021
Accepted: 13 December 2021
Published: 17 December 2021

1. Introduction

Text has become one of the essential means of conveying information in the contemporary world, and there is a wide variety of textual information in the social scenes we live in. Detecting the text in the natural environment is the process of locating text regions in an image through a detection network and representing them with polygonal bounding boxes. Accurate detection results are beneficial to comprehensive practical applications, such as instant translation, image retrieval, scene analysis, geographic location, license plate recognition, and so forth, which has aroused strong interest in the domain of computer vision and document analysis. Existing CNN-based text detection algorithms [1,2] can be divided into approximately two categories: regression-based and segmentation methods.

For regression-based scene text detection algorithms [3–12], text objects are usually represented in the form of a rectangular or square field with a certain orientation. Although the detection speed is fast and can avoid the generation of errors that accumulate over multiple stages, most existing relapsing-based ways are no longer able to handle the text detection problem accurately and efficiently due to the limited form of the text representation (axis-aligned rectangles, rotated rectangles or quadrilaterals), and in particular do not perform very well when used to detect curved text on datasets such as Total-Text [13], which is very unfavorable to the subsequent text recognition in the whole optical character recognition engine.

On the other hand, segmentation-based scene text detection algorithms [14–20] focus on locating text instances by classifying pixels. Although recent approaches have made telling improvements in scene text detection tasks, the focus has shifted from lateral text to multidirectional text and more sophisticated forms of text (curved text). There are two other challenges that need to be faced when detecting arbitrarily shaped scene text due to specific properties of the scene text, such as significant variations in color, scale, orientation, proportion, and shape that clearly distinguish it from the common target object, along with various properties of natural pictures, such as degree of image blur, intensity of illumination and so forth.

The first challenge is to refine features. First of all, the network formed by stacking standard convolutional [21,22] layers lacks sufficient high-semantic information extraction and storage capabilities. Specifically, the entire network learns through similar patterns, and the feature information extraction is incomplete. Secondly, under the complex background conditions, because of the limitation of CNN receptive field size, the text information in the image cannot be effectively utilized, thus it is impossible to arbitrarily localize the text more accurately. Therefore, to solve the above problems, we introduce two modules—the depth weighted convolution module and the 3D-Attention module. The property of scene text detection is improved by increasing the depth of the CNN model, with more parameters and a deeper network to learn more complex feature information.

The second challenge is the large-scale variation in the scene text. Firstly, the scale variation in scene text is much larger than that of a general target object, which makes it hard for CNN to learn a specimen. Secondly, as different scale feature layers have different distribution characteristics, the deep feature layer has rich semantic information but lacks accurate location information, while the shallow feature layer has detailed and rich information but introduces a large amount of redundant information, which can make some of the regions to belong to inappropriate areas (e.g., non-text regions) to be classified [23] incorrectly. Therefore, to settle this question, a new cross-level feature pyramid network is proposed in this paper to obtain feature maps of text representation at various scales. By aggregating these multi-scale feature maps, the problem of large-scale differences in scene texts is effectively solved, and the text area is located more finely, and it can be easily utilized in existing methods.

This article proposes a new text detector to effectively solve these two problems, allowing a more accurate detection of the text of the scene in any form. As shown in Figure 1, obtain an input image, then the feature pyramid backbone ResNeDt generates layers of different scales through a downsampling operation. Compared with the original residual network, ResNeDt increases the depth of the network, further enlarges the receptive field and adapts more effectively to arbitrarily shaped text, that is, horizontal, multi-directional, curved and wavy texts, thus achieving the finer localization of text regions. To collect the feature information for the surface layer and the deep layer comprehensively at the same time, we propose the Cross-Level Feature Pyramid Network for modelling the extracted feature information on two adjacent feature layers to further enhance feature extraction. To take advantage of this property, the module can produce multi-scale feature representations, effectively solving the problem of multi-scale variation in scene text detection with minimal increase in computational effort. Finally, the binarization map is obtained through adaptive learning of the differentiable binarization module to produce higher quality prediction boxes, further improving the robustness of text detection for various shapes.

To demonstrate the validity of our proposed framework, experiments have been carried out on three different types of datasets, containing ICDAR 2015 [24], Total-Text [13] and MSRA-TD500 [25]. In these datasets, Total-Text is specifically designed for curve text detection. Therefore, experimental results in the MSRA-TD500 and Total-Text datasets show that this method has high flexibility in complex situations (such as multilingual text, curved text, arbitrarily shaped text). Specifically, on Total-Text with arbitrarily shaped text, it significantly exceeded the results of most of the most modern methods, and our model

achieves a comparable performance (82.6%). In addition, the framework proposed in this paper also achieves a good performance on the ICDAR 2015.

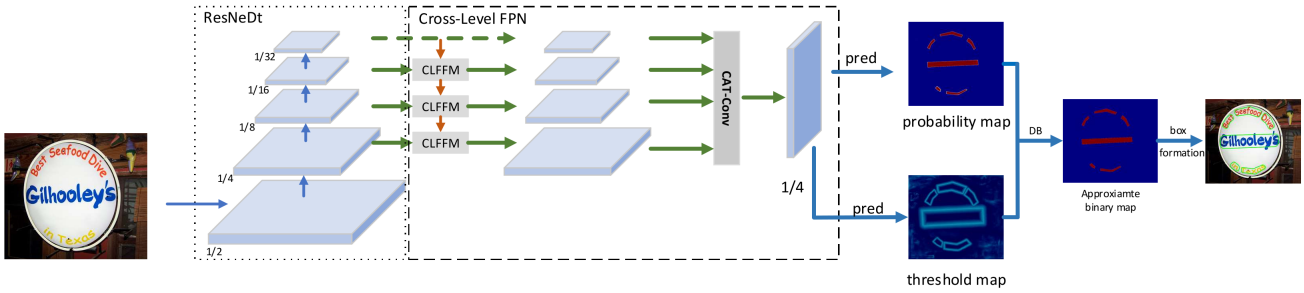

Figure 1. The general architecture of the proposed CSFF-Net. ResNeDt is considered to be the backbone network. Our proposed method is mainly processed in three steps: Firstly, the pictures are output to different feature layers through the backbone network ResNeDt. Secondly, the outputs of the backbone network extract detailed feature information by the Cross-Level FPN. At last, the output result of the Cross-Level FPN is obtained by the DB module.

To sum up, the primary contributions of this article are as follows: (1) a deep weighted convolution module is proposed to produce more expressive features, which is a more efficient method with a universal structure and less computational effort than previous methods; (2) The proposed 3D-Attention module can model the contextual relevance of characteristic graph, thus improving the performance of text detection, and generating more representative features; (3) Cross-Level Feature Pyramid Network with Cross-Level Feature Fusion Module, which not only handles feedforward information flow efficiently but also enriches features through upper (lower) feature layers as well as jump connections, can effectively solve the problem of detecting arbitrarily shaped scene text, and improve the performance of text detection; (4) This article has realized the most advanced performance on several benchmarks including different forms of text instances (oriented, long, multilingual and curved), which proves the superiority of our newly designed module.

2. Related Work

The detection of scene texts has been a popular research theme and many means have been proposed. Before the appearance of deep learning, early text detectors [26–28] mainly used hand-crafted features as basic components, such as Stroke Width Transform (STW) [26], Maximally Stable Extremal Regions (MSER) [27] and symmetry feature [28]. In recent years, scene text detection method based on deep learning have achieved remarkable effects. Modern text detectors are mainly based on CNN and these methods can be divided into two categories, that is, methods based on regression and segmentation.

Regression-based detection methods typically follow a target detection framework driven by convolutional neural networks (CNNs) [1,2], such as Faster R-CNN and SSD [29]. Unlike ordinary objects, text is usually displayed in irregular shapes with different proportions. To deal with this problem, TextBoxes [3] used SSD as the base detector, modifying the size and shape of the convolutional kernel anchor box to accommodate variations in the proportion of text instances. As versions of the Faster RCNN, the Rotation Region Proposal Network (RRPN) [4] and Rotational Region CNN (R2CNN) [1] were designed to detect text in arbitrary directions in a two-stage manner. To handle the detection of long text, Baoguang et al. [5] and Zhi et al. [6] proposed SegLink and CTPN, which predicted text fragments and connected them into text boxes. RRD [7] extracted feature maps from two separate branches for text classification and regression to better detect long texts. Reference [8] obtained text vertices and grouped them into boxes. Unlike these methods, which regress anchor boxes/segments/corners, Xinyu et al. [9] and Wenhao et al. [10] performed box regression and predicted pixel offsets in the text area without using anchors and suggestions. Chuhui et al. [11] based on [9], the boundary region of the text is divided to distinguish the text instances. However, there are certain structural limitations in using

this method to capture all possible shapes. The newly proposed LOMO [12] suggests an initiative refinement module to bitterly refine bounding box proposals for extremely long text, and then provides for centerline offsets, text region, and frame offsets to recreate the text instance.

Although regression-based approaches have achieved advanced performance, which still require tedious multi-stage tasks, which might require comprehensive adjustment and lead to sub-optimal performance. In addition, due to the huge difference in the aspect ratio of texts (especially non-Latin texts) and the limited receiving domain of CNN, these methods cannot effectively deal with texts under complex background conditions.

Segmentation-based approaches [30,31] mainly draw on semantic segmentation methods, where all pixels in a text bounding box are considered positive sample regions, and describe the text areas by adopting different representations, and then reconstruct the text instances through specific post-processing. Yongmin et al. [14] put forward the method of character probability prediction. The main idea of this algorithm is to use a Gaussian heat map to generate a heat map of a single character, and then use the distance between characters to generate an affinity heat map for weak supervision training and learning. This method is effective for dealing with languages with constant character spacing. Reference [15] formulated a range of text as multiple attributes, such as text region and orientation, and predicted the corresponding heatmaps by FCN to extract the text region. Liu et al. [16] proposed a Transverse and Longitudinal Offset Connection (TLOC) based on [32] and RNNs to directly regress the polygon shape of textboxes. Reference [17] considers the detection of curved text as an instance segmentation problem and uses MASK RCNN to generate the boundaries of text instances. The component segmentation method divides the text area into several components that are grouped into different instances by grouping data, communicating between nodes, or post-processing. For example, PixelLink [18] predicts connections between pixels and finds text area and separates links belonging to different text instances. Tian et al. [19] designed a two stages method to separate dense text instances. PSENet [20] is gradually extending kernels at a certain scale to split nearby text instances. Our method combines the advantages of goal detection and segmentation methods, adopts a three-step structure, and uses contour points to represent text areas. This model effectively solves the problem of large-scale differences and enhances the text-related regions by reducing the background interference of the feature layer and the use of the attention mechanism. Compared with the previous methods, this method gives a more accurate description of the text regions, so it can produce finer text boundaries.

3. Methods

Deep convolutional neural networks [21,22] have made a series of breakthroughs in image classification [21,23,33] and are able to effectively learn and understand high-level semantic information directly from visual images because of their powerful feature representation capabilities. In order to improve network performance, build lightweight networks that are easy to deploy and meet the requirements for real-time performance in practical applications, in this paper we chose DBNet [34] as our baseline and improved the original neural network. Without adjusting the model infrastructure and ensuring the original feature information of the network, it improves the feature expression process of the backbone network, and introduces a depth weighted convolution module (DWCM) and a 3D-Attention module to model the context relevance of effective features, further optimizing the feature extraction network and enhancing the effectiveness. As shown in Figure 2, not only can the depth of the network be increased compared with the original network, but also the detection accuracy of the model is improved.

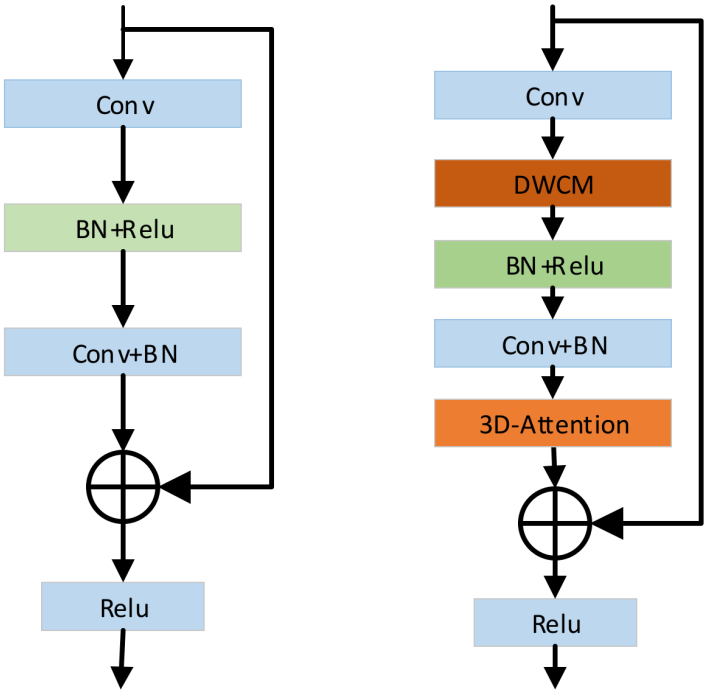

Figure 2. The left figure represents the residual structure of the original paper and the right figure represents the residual structure of this paper. We further extract feature information by embedding DWCM and 3D-Attention.

Our overall network structure is shown in Figure 1, and its specific steps and roles are shown in Table 1. It can be seen from Table 1 that the network is mainly composed of three parts, namely Backbone, Neck and Head. Backbone refers to ResNeDt18 and ResNeDt50 in this paper, and its role is to extract the feature information in the image for later network use. Neck is placed between Backbone and Head. The Neck in this paper is our Cross-level FPN, which can make better use of the features extracted by the Backbone to generate more representative features. The Head is the detection head, which is the network that acquires the output content of the network. The Head here is Differentiable Binarization, which predicts the text boxes by using the features extracted before.

Table 1. The steps and roles of our model.

	Module		Input	Output	Roles
Backbone	ResNeDt(18/50)		I_{in}	C_2, C_3, C_4, P_5	Generating multi-level features
Neck	Coss-level FPN	CLFFM	P_5, C_4	P_4', P_4	Generating correction and output values
		CLFFM	P_4', C_3	P_3', P_3	Generating correction and output values
		CLFFM	P_3', C_2	_ , P_2	Generating output values
Head	Differentiable Binarization		Probability map Threshold map	Binary map	Generating prediction box

3.1. Depth Weighted Convolution Module (DWCM)

For any series of residual networks (e.g., ResNet 18, 34, 50, 101, 152), the structure of the front part is the same—7 × 7 standard convolutional layers and 3 × 3 maximum pooling layers—and then a series of respective residual structures formed by stacking several standard convolution layers. For standard convolution, the output feature mapping for the i-th channel can be expressed as follows:

$$y_i = k_i * X = \sum_{j=1}^{C} k_i^j * X_j, \quad (1)$$

where * denotes the convolution handle and k_i is the convolution kernel size.

However, the standard convolution has the following shortcomings: each standard convolutional output feature map must sum all channels, and all the feature layers of the original network are repeatedly generated by Equation (1). It is known that the entire network is learned using similar patterns. In addition, the network stacked by the standard convolution layers also lacks enough high semantic information storage capacity and cannot capture high semantic information. In addition, the presence of pooling layers somewhat causes insensitivity to image details, and the content of target information is less under complex background conditions, which leads to the inability to accurately locate the target object. As a result, the above drawbacks may result in a weaker representation of the feature map. To alleviate the above problems and improve the detection performance of convolutional networks, we specifically designed a novel convolutional neural network structure, as shown in Figure 3, which is added between the standard convolution layers to learn more image features.

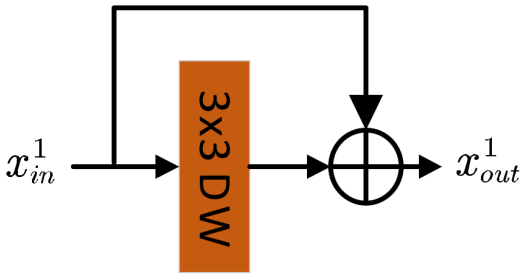

Figure 3. Design of the Deep Weighted Convolution Module (DWCM).

As an enhanced version of standard convolution, our depth weighted convolution module consists of two main steps: firstly, a 3 × 3 convolution operation is performed independently on each channel of the input, and then the output features are summed with the input features of the module element by element, that is, the ⊕ operation. The advantage of such processing is that it retains the reuse of feature information from the original network, reduces the loss of low-dimensional feature information and ensures that the network learns richer features at each spatial dimension. This process can be expressed by mathematical Formula (2):

$$x_{out}^1 = f_{DWCM}\left(x_{in}^1\right) \oplus x_{in}^1 \tag{2}$$

where f_{DWCM} represents the 3 × 3 convolutional layer, x_{in}^1 and x_{out}^1 represents input and output, respectively.

Here, the single-channel convolution operation is performed on a two-dimensional plane and a single convolution kernel is applied to each channel. For example, the sample input size is set to H (image height) × W (image width) × K (number of image channels). The 3 × 3 convolution is chosen here because the 3 × 3 convolution structure is more computationally intensive in the GPU than the 1 × 1 convolution or even the 5 × 5 convolution, and using the 3 × 3 convolution structure is faster in the GPU operation. Each channel of the input feature is convolved with the corresponding convolution kernel of a single channel, so that the number of feature maps is kept unchanged. Here, after K-channel convolution operation, K feature maps ($H \times W$) are still obtained, so the purpose of filtering the input features can be achieved, providing more efficient input features for subsequent operations. The expression is as follows:

$$G_{i,j,m} = \sum_{w,h}^{W,H} K_{w,h,m} \odot X_{i+w,j+h,m} \tag{3}$$

In Equation (3), G is the output, K is the convolution kernel of width (W) and height (H), X is the input, m denotes the m-th channel of the feature map, i,j denotes the coordinates of

the output on the m-th channel, and w and h are the coordinates of the convolution kernel weight elements of the channel.

Compared with the standard convolution operation, this module has two advantages. Firstly, it enhances the information of the channels, making the feature representation generated by our depth-weighted convolutional module more expressive and making the network model achieve better results. As a result, the residual network with the depth weighted convolutional module can locate text regions more completely and more accurately. Secondly, the depth weighted convolution module is universal and can be easily applied to standard convolutional layers without introducing any parameters or changing hyperparameters, thus achieving portability.

3.2. 3D-Attention Module

It is well known that human visual processing ability is limited and cannot process all the information at the same time. Attention is mainly focused on regions with significant features, and machine vision can also use this attention mechanism to effectively improve work efficiency. There is a large amount of information in complex scenes, and the most important information in an image is generally concentrated in a relatively small area, so using visual attention mechanisms to quickly and accurately acquire effective information from an image is particularly important in the process of visual model building. Therefore, inspired by this, we designed a simple and universal 3D-Attention module that applies it to the features in each BasicBlock [35] together with training, aiming to extract effective features to suppress ineffective features and screen high semantic feature information, enhance the network's ability to refine features and make the network more focused on information features, such as text regions in images, which can effectively improve the network's feature extraction ability and increase the model's expressiveness.

The module significantly expands the receptive field of each feature layer by improving the feature transformation of the convolutional network, helping the CNN to produce more representative information, enhancing the learning representation of the network, enriching the output features of the backbone network and improving the accuracy of feature extraction, thus further optimizing the network. Compared with other attention mechanisms, no additional parameters are introduced and only a small amount of computation is added to improve the model performance with a smaller overhead. First, we briefly introduce the channel attention mechanism [36] and the spatial attention mechanism [32]. The purpose of [36] is to obtain a one-dimensional feature vector with a size of (C $\times$ 1 $\times$ 1), while spatial attention obtains a two-dimensional feature map with a size of (1 $\times$ H $\times$ W). It is worth noting that C denotes channels number, and H and W are the height and width of the characteristic graph, respectively. The 3D-Attention module in this paper is similar to Zhu et al. [32] and Hu et al. [36], but there are some differences. The difference is that our attention produces a three-dimensional matrix (C $\times$ H $\times$ W) as an attention feature map, rather than a one-dimensional feature vector or a two-dimensional feature map. As shown in Figure 4.

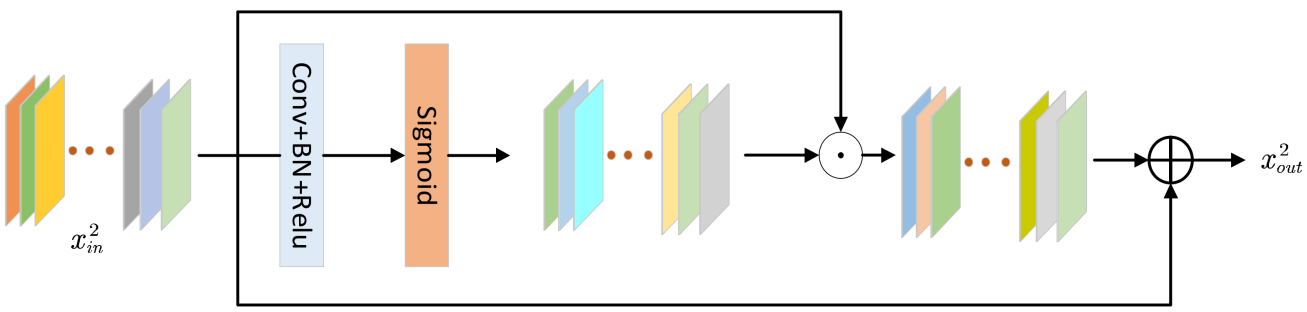

Figure 4. Architecture of the 3D-Attention module.

The 3D-Attention module uses only a standard ConvBnRelu (1×1 convolutional) and sigmoid activation function to obtain the attention feature map, and then multiplies the attention feature map by the input of the module and then adds it with the input features, thus obtaining a high semantic feature map under the 3D-Attention module, introducing fewer additional parameters to enhance the sensitivity of the network to text, and generate better detection results. The input of this module is the feature map output by the previous convolution block, and the attention module provides the position index corresponding to its dimension. We denote the input feature map as x_{in}^2 and the output feature map as x_{out}^2, with x_{out}^2 passed on to the next stage as the module output. Thus, the 3D-Attention module can be described as follows.

$$x_{out}^2 = f_{3D}\left(x_{in}^2\right) \odot x_{in}^2 \oplus x_{in}^2, \tag{4}$$

in which f_{3D} represents the 1×1 convolution layer, batch normalization layer BN and nonlinear layer $Relu$, followed by a sigmoid. The BN can prevent data distribution from shifting after matrix multiplication and nonlinear operation, which will slow down the convergence of network. Passing through the BN layer effectively avoids the gradient disappearance and explosion problems of deep networks, and also reduces the reliance on parameter initialization methods. The Sigmoid function lies in the output of a probability map that determines the weights of each feature. The non-linear relationship between the channels is constructed using the $Relu$ activation function and the sigmoid function to enhance the non-linear capabilities of the model, improve the learning representation of the network. It is worth noting here that we have placed the 3D-Attention module after the depth weighted convolution module (DWCM), and only in this way can maximize the usefulness of each module.

The 3D-Attention module proposed in this paper not only calibrates the features between channels, but also improves the local feature representation of spatial domain information. In the process of calibration, spatial features and channel information are effectively combined to further enrich the contextual semantic information of small targets (text) in shallow features. The advantages are mainly reflected in the following three aspects:

Firstly, compared to standard convolution, each spatial location not only embeds its surrounding information as a scalar of the original spatial response, but also models the long-distance inter-channel dependencies to capture the rich contextual relationships. Screening each input channel facilitates the network to selectively enhance features containing useful information and suppress redundant features, thereby effectively improving the transferability of target features between high and low layers and enhancing the semantic information in the feature layer.

Secondly, instead of collecting global contextual information, the 3D-Attention module only considers the contextual information around each spatial location, which to some extent avoids certain pollution information from irrelevant regions (non-text). It also uses a residual connection structure in the deeper part of the network to further enhance the information transfer between non-adjacent feature layers, improve feature utilization, avoid the gradient disappearance problem and make the network layers deeper.

Finally, the 3D-Attention module can be easily embedded into modern classification networks for a wide range of tasks due to its generic nature. Although it increases the number of parameters in the network, the structure is simple. The introduction into existing networks will only add a small amount of computation and model complexity, with good generalization to different datasets, which is extremely attractive.

3.3. Cross-Level Feature Pyramid Networks

At present, many networks only use a single high-level feature to classify objects, but there is an obvious defect, that is, small objects (such as text) have less pixel information and are easily lost during the up-sampling process. In view of this kind of object size is different from the general object detection, the classic approach is to enhance multi-scale changes by using image pyramids, but this will bring a great deal of computation. Therefore, this paper

proposes a Cross-Level Feature Pyramid Network (Cross-Level FPN) based on Feature Pyramid Network (FPN), as shown in Figure 5.

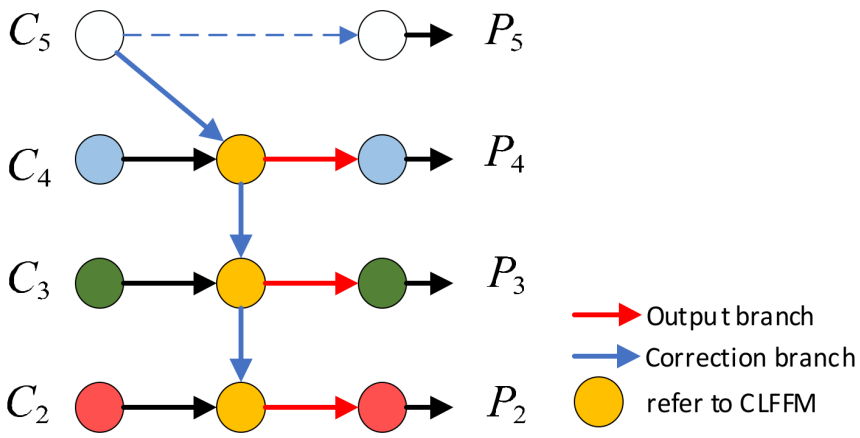

Figure 5. Design of a Cross-Level Feature Pyramid Network.

Cross-Level FPN is a top-down network structure with horizontal connections, which is used to construct feature maps with high semantic information of different sizes. Specifically, the inputs $\{C_2, C_3, C_4, C_5\}$ are the outputs of the backbone (ResNeDt), their sizes are $\left\{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\right\}$ of the original size, corresponding to the outputs of level 2, 3, 4, 5 respectively. The level refers to each stage of the network. In general, the output feature maps that produce the same size are considered to be at the same level, and each level is defined as a stage, and the output of the last stage of each stage serves as the input of Cross-Level FPN, which enables us to create a pyramid network that contains more semantic information. Level 1 is not included in the feature pyramid network as it is too large in size and takes up a lot of memory.

A multi-scale feature representation is generated by extracting features for each scale of the image. Information from both high-resolution lower features and high-semantics higher features are used to predict the feature maps at each level. It can effectively solve the multi-scale variation in scene text detection with minimal computation. Thus, it can be concluded that shallow feature layers (such as C_2) contain more textured (detail) information, while deeper feature maps (such as C_5) contain more semantic information. In order to combine feature maps with different features, Cross-Level FPN uses top-down and lateral linking strategies. The top-down path produces higher resolution features by upsampling that are smaller in spatial size but more semantically informative at higher pyramidal network levels. Then, the features are further enhanced by transverse connection. It should be noted that the feature sizes of transverse connections are the same. As shown in Figure 5, the red arrows represent the output branches, the blue arrows represent the correction branches, and the yellow circles indicate the Cross-Level Feature Fusion Module (CLFFM). The correction branch of feature layer C_5 is corrected by CLFFM for feature layer C_4 to obtain the output branch and correction branch of C_4; the correction branch of feature layer C_4 is corrected by CLFFM for feature layer C_3 to obtain the output branch and correction branch of C_3; the correction branch of feature layer C_3 is corrected by CLFFM for feature layer C_2 to obtain the output branch of C_2 branch. The output branches of all feature layers are fed into the next stage of the task as the output of the Cross-Level FPN.

The working mechanism of CLFFM is mainly introduced by taking two input feature layers C_4 and C_5 as examples, as shown in Figure 6.

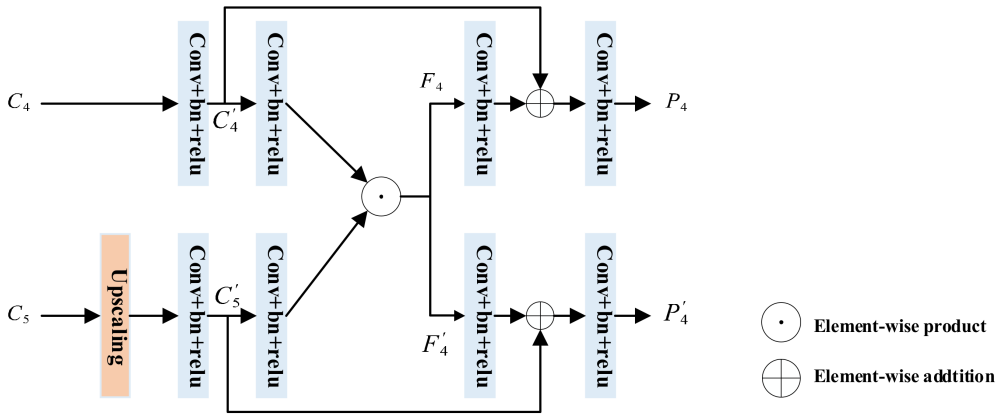

Figure 6. Architecture of the Cross-Level Feature Fusion Module (CLFFM).

First of all, for the smaller resolution feature map (C_5) the bilinear interpolation method is used to improve the resolution to the same scale as (C_4), and cross-layer feature maps C_4' and C_5' on their branches are generated by a convolution operation:

$$C_4' = Relu(Bn(Conv(C_4))) \tag{5}$$

$$C_5' = Relu\left(Bn\left(Conv\left(C_5^{\uparrow}\right)\right)\right) \tag{6}$$

where $C_5^{\uparrow}$ denotes the upsampling operation, $Conv$ is the 3×3 convolution, Bn denotes normalisation and $Relu$ denotes the activation function.

Secondly, we multiply the generated cross-layer feature maps C_4' and C_5' by convolution operation and element by element, and output two branches, output branch F_4 and correction branch F_4':

$$F_4 = Relu\left(Bn\left(Conv\left(C_4'\right)\right)\right) \odot Relu\left(Bn\left(Conv\left(C_5'\right)\right)\right) \tag{7}$$

$$F_4' = Relu\left(Bn\left(Conv\left(C_4'\right)\right)\right) \odot Relu\left(Bn\left(Conv\left(C_5'\right)\right)\right) \tag{8}$$

where $\odot$ represents dot multiplication operation. Here, the processed two-level features are point multiplied. The purpose of this is that lower-level features can provide more accurate location information, while the up-sampling operation will cause errors in the positioning information of the deep network, so we combine them to form a deeper feature pyramid network, which integrates multiple layers of feature information and outputs them in different features.

Then, both F_4 on the output branch and F_4' on the correction branch are passed through a convolution with a channel number of 64 and a convolution kernel size of 3×3. The resulting feature maps are then summed element by element with the cross-layer feature maps C_4' and C_5' respectively for feature fusion, and finally a 3×3 convolution is appended to generate the final feature maps P_4 and P_4'.

$$P_4 = Relu\left(Bn\left(Conv\left(\left[Relu(Bn(Conv(F_4))) \oplus C_4'\right]\right)\right)\right) \tag{9}$$

$$P_4' = Relu\left(Bn\left(Conv\left(\left[Relu\left(Bn\left(Conv\left(F_4'\right)\right)\right) \oplus C_5'\right]\right)\right)\right) \tag{10}$$

where $\oplus$ denotes the element-by-element summation operation.

Finally, the purpose of convolution operation is to reduce the confounding effect caused by upsampling and further ensure the integrity of pyramid network structure. The reason for outputting two branches is, on the one hand, because P_4' on the correction branch can be used as an input to repeat the above process with the feature map C_3 of the previous stage. In this way, the high semantic information of the deep feature map is retained, which can be perfectly fused with the low-level feature map to further enhance feature extraction. On the other hand, one of the outputs of the CLFFM module (P_4), that combines

the high semantic information of the high-level feature with the rich details of the low-level feature, thus obtaining the final highly accurate feature map, is called $\{P_2, P_3, P_4, P_5\}$, which corresponds to the input feature map with the same size $\{C_2, C_3, C_4, C_5\}$. This process can be expressed in mathematical Equation (11) as follows.

$$P_i, P_{i+1}, \ldots, P_{i+n} = f(C_i, C_{i+1}, \ldots, C_{i+n}) \tag{11}$$

At the beginning of the iteration, it is necessary to add a 1×1 convolution to each input feature layer $\{C_2, C_3, C_4, C_5\}$ to reduce the dimension, so as to ensure the consistency of the number of channels.

3.4. Differentiable Binarization

The structure of differentiable binarization is shown in Figure 7. The input is an image with text, and after the network a segmentation probability map P is obtained along with an adaptive threshold map T (each pixel on the image has a corresponding threshold and each pixel has a different threshold). The final result is obtained by performing a differentiable binarization operation using the P and T outputs. Specifically, after the enhanced feature extraction network Cross-Level FPN outputs four feature layers $\{P_2, P_3, P_4, P_5\}$, three of the feature layers $\{P_3, P_4, P_5\}$ are upsampled to the largest size feature layer P_2. Then these four feature layers are spliced together to obtain a feature layer F, which has the same size as P_2. The F is used to predict P and T. Finally, P and T are combined to obtain the binarized map $\hat{B}$.

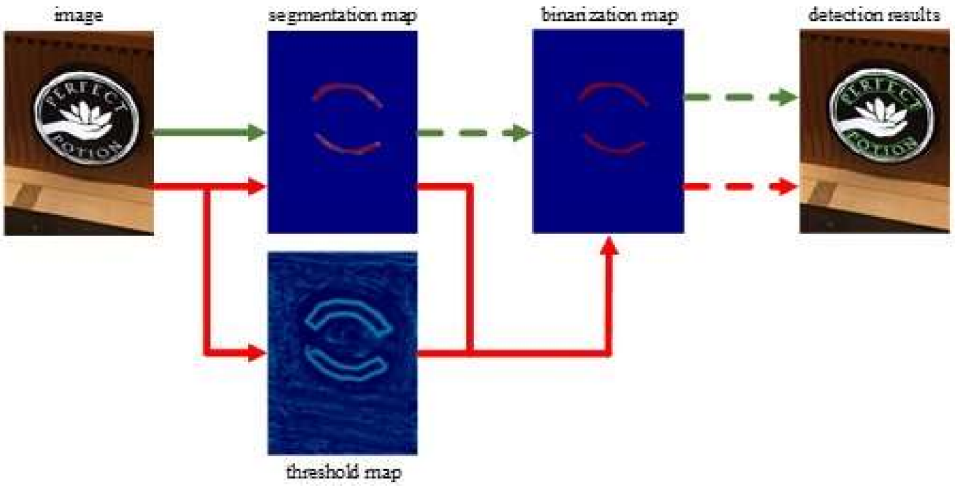

Figure 7. The blue paths represent the standard binarization process and the dashed lines only represent the inference process; the red paths are the differentiable binarization used in this paper, which adaptively predicts the threshold at each position of the image.

4. Results

4.1. Datasets

In this paper, experiments are carried out on three challenging public datasets. They are Total-Text [13], MSRA-TD500 [25], ICDAR15 [24]. The visualization results are shown in Figure 8.

Figure 8. Visualization results of our method on different types of text examples, including curved, multidirectional, multilingual texts. For each unit in the above diagram, the probability diagram is in the upper right corner, the threshold map is in the lower right corner.

1. Total-Text [13] is a dataset used for detecting curved texts, which contains the curved texts of commercial signs and sign entrances in real-life scenes, with a total of 1555 pictures, 1255 training sets, and 300 the test sets.
2. MSRA-TD500 [25] belongs to a multi-language and multi-category dataset, with 500 photos, 300 for training, and 200 for testing. These photos are used to shoot signs, house numbers and warning signs in indoor scenes and guide plates, and billboards in some complex backgrounds in outdoor sets.
3. ICDAR2015 [24] is a linear detection and recognition dataset belonging to the English class, with 1500 images, including 1000 training pictures and 500 test pictures. This dataset is a street or shopping mall image taken randomly by Google Glass without focusing; the goal is to improve the generalization performance of the detection model.

4.2. Loss Functions

The loss function plays a crucial role in deep neural networks, the L_1 loss function and the binary cross-entropy loss function are used to optimize our network. The loss function in this paper consists of three components in the training process: probability map loss L_s, binarization map loss L_b, and adaptive threshold map loss L_t, represented as follows:

$$L = L_S + \alpha \times L_b + \beta \times L_t, \tag{12}$$

where α and β are the weight parameters, α is set to 1 and β is set to 10.

Among them, the binary cross-entropy loss function is used for probability map loss L_s and binary map loss L_b. The formula is as follows, and negative hard mining [34] is used to overcome the imbalance between positive and negative samples.

$$L_s = L_b = \sum_{i \in S_l} y_i log x_i + (1 - y_i) log(1 - x_i) \tag{13}$$

in which S_l represents samples whose positive and negative ratio is 1:3, and L_1 loss function is adopted for the loss L_t of the adaptive threshold map, and its formula is:

$$L_t = \sum_{i \in R_d} |y_i^* - x_i^*|, \tag{14}$$

where R_d is the index of the pixels in the region and y^* is the label of the adaptive threshold map.

4.3. Implementation Details

The experiments in this paper use Python 3.7 as the programming language and the deep learning framework used is Pytorch1.5. All the experiments were conducted on TITAN RTX. The initial learning rate was set to 0.007. The training process involved two

steps: firstly, the network was trained for 100k iterations by using the SynthText dataset [37], then the models were finetuned on the benchmark real-world datasets for 1200 epochs. Our model was trained by using the official training image of each dataset, a weight decay of 0.0001 and a momentum of 0.9. The training optimizer was Adam [38], and the training batch size was set to 16. The text marked as "NEGLECT" was discarded in the training process. In the pre-processing stage of the network, the labels of the probability map and threshold map were created based on the labels of the train dataset. Since smaller text regions are not easy to detect, some text regions that were too small (e.g., Minimum side length of the smallest rectangle of the text area less than three or polygon area less than 1 were ignored in the process of creating the labels. As a result, small regions of text marked as "NEGLECT" were discarded during the training process. In the testing stage, single-scale images were input, and the results were evaluated by the official evaluation protocol.

Because the test images of different scales have great influence on the detection effect [6,8], the aspect ratio of the test images was kept in the reasoning stage, and the size of the input image was adjusted by setting a suitable height for each dataset.

We made full use of and expanded the data in the same way as in [34], mainly in the following three ways: (1) random rotation; (2) random cutting; (3) random flipping. In order to improve the training efficiency, the processed images were all adjusted to 640×640.

4.4. Ablation Study

In order to better prove the realization of each module proposed in this paper, ablation research was carried out, which proved the effectiveness of our proposed Deep Weighted Convolution Module (DWCM), the 3D-Attention Module, and the Cross-Level Feature Pyramid Network (Cross-Level FPN). In the ablation experiments, we tested DBNet, DBNet+DWCM,DBNet+3D-Attention,DBNet+DWCM+3D-Attention,DBNet+Cross-Level FPN and the method proposed in this paper (DBNet+DWCM+3D-Attention+Cross-Level FPN). The detailed experimental results are shown in Table 2. It can be seen from Table 2 that the precision, recall and F-measure of the baseline DBNet on the test dataset ICDAR2015 are 89.3%, 73.8% and 80.8%, respectively. Our method DBNet+DWCM+3D-Attention+Cross-Level FPN has a precision, recall and F-measure of 86.4%, 79.2% and 82.7%, respectively. The F-measure of this method on ICDAR2015 is 1.9% higher than the baseline, and the detection result is obviously better than the original DBNet. We explored the performance of the proposed module on baseline through ablation experiments, the results of which are shown in Table 2. Table 2 shows the impact of the different modules on the performance of the network, with the final network DBNet+DWCM+3D-Attention+Cross-Level FPN achieving a better performance; thus, proving the validity of our proposed module. It is worth noting that DBNet is our baseline.

Table 2. Test results under different settings. "P", "R" and "F" respectively represent precision, recall and F-measure.

Backbone	DWCM	3D-Attention	Cross-Level FPN	P	R	F
Resnet-18				89.3	73.8	80.8
Resnet-18	√			86.9	76.0	81.1
Resnet-18		√		87.5	76.2	81.5
Resnet-18	√	√		87.6	77.7	82.3
Resnet-18			√	88.6	76.3	82.0
Resnet-18	√	√	√	86.4	79.2	82.7

Figure 9 shows the visualization results of GT, baseline and our method, respectively. It is worth noting that the images in the figure (from top to bottom) are randomly selected from the test datasets ICDAR2015, MSRA-TD500 and Total-Text. The images here are randomly selected from three datasets, which can better prove the robustness of our model.

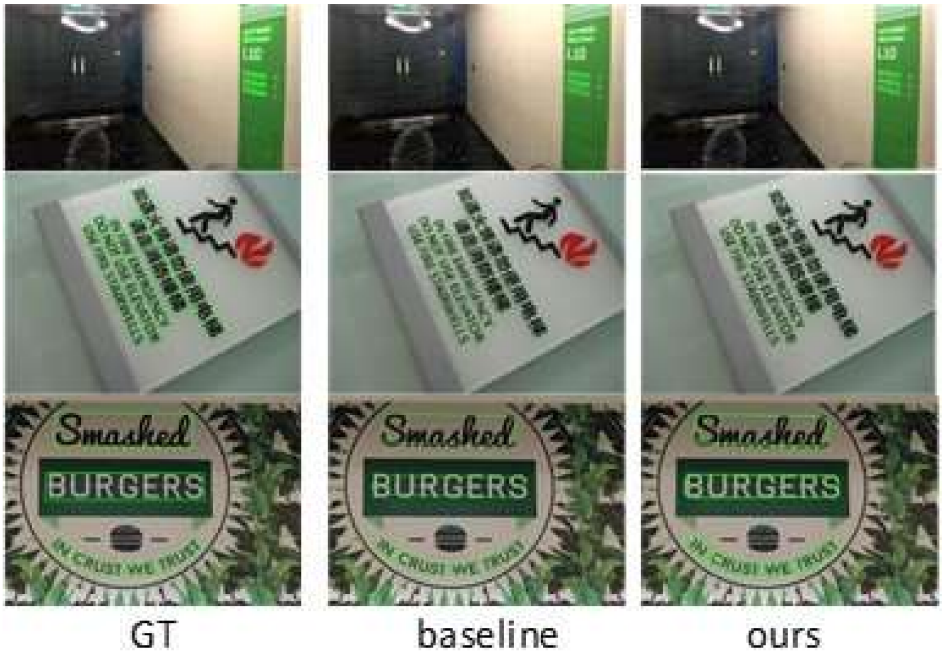

Figure 9. Visualization results of our method. After a series of enhanced feature extraction operations, the position information of the text box is more accurate and our method (ours) is more effective than baseline.

4.4.1. 3D-Attention

DBNet+3D-Attention can effectively remove some irrelevant information and make the prediction box closer to the real box. In Table 1, it can be seen that the 3D-Attention module significantly improves the performance of ResNet-18. Specifically, using the ResNet-18 backbone network, the F-measure of the 3D-Attention module on the ICDAR2015 dataset has been improved by 0.7%, and the recall has been improved by 2.4%.

4.4.2. DWCM

Compared to DBNet, the DBNet+DWCM method will yield richer features when the DWCM is added. As shown in Table 1, the deep weighted convolution module can also result in a performance gain of 0.3% as it extends the receptive domain of the backbone network, it takes only a little extra time. For ResNet-18, depth weighted convolution improves the recall rate by 2.2% on the ICDAR2015 dataset.

4.4.3. 3D-Attention+DWCM

Faced with different types of complex datasets, we take advantage of two modules, the Deep Weighted Convolution and the 3D-Attention module, as a starting point and propose DBNet+DWCM+3D-Attention to meet the challenges posed by this complexity. From Table 1 we can know that, for the ICDAR2015 dataset, 1.5% (with ResNet-18) improvements are achieved by the 3D-Attention and the DWCM, the precision of 87.6%, recall of 77.7%, F-measure of 82.3%. Compared to DBNet+DWCM and DBNet+3D-Attention and DBNet+DWCM+3D-Attention achieves 1.2% and 0.8% performance gain in terms of F-measure, respectively. Thus, it can be seen that the detection performance of a network combining the advantages of these two modules is better than that of a network applying either the depth weighted convolution module or the 3D-Attention module alone.

4.4.4. Cross-Level FPN

DBNet+Cross-Level FPN is able to fully capture text areas through constant supplementation and fusion when dealing with irregularly shaped text in complex background conditions, with better detection results than DBNet. As can be seen from Table 2, with the help of Cross-Level Feature Pyramid Networks, due to increased network representa-

tion capabilities, the proposed method achieves a result of 76.3%, 88.6%, 82.0% in recall, precision and F-measure, respectively; it can bring a performance gain of 1.2%. The experimental results illustrate that cross-level feature pyramid networks can indeed extract feature information more comprehensively and improve image classification accuracy. It has stronger feature capture capability than the original feature pyramid network (FPN).

4.5. Compare with Previous Method

We compare our proposed method with previous methods on different datasets, including a benchmark for curved text, a benchmark for multi-directional text, and a benchmark for long text and multiple languages. Based on the evaluation criteria proposed in Herbert et al. [39] and Mark et al. [40], the experimental results are reported in Tables 3–5. Compared with the basic network DBNet, our approach shows significant improvements on all three datasets. Especially on the Total-Text dataset on the Total-Text dataset, the method of this paper also shows a corresponding improvement in each metric compared to the base network. Similarly, for the MSRA-TD500 dataset, the method outperforms its competitors in terms of P, R, F. By comparing the P, R, F on three datasets, our proposed module is robust in terms of improving text detection performance.

Table 3. Test results on curve datasets. The values in brackets refer to the height of the input image. "*" refers to multi-scale test. "MTS" and "PSE" are short for Mask TextSpotter and PSENet.

Method	P (%)	R (%)	F (%)
TextSnake (Long et al., 2018)	82.7	74.5	78.4
ATRR (Wang et al., 2019b)	80.9	76.2	78.5
MTS (Lyu et al., 2018a)	82.5	75.6	78.6
TextField (Xu et al., 2019)	81.2	79.9	80.6
LOMO (Zhang et al., 2019) *	87.6	79.3	83.3
CRAFT (Baek et al., 2019)	87.6	79.9	83.6
CSE (Liu et al., 2019b)	81.4	79.1	80.2
PSE-1s (Wang et al., 2019a)	84.0	78.0	80.9
DB-ResNet-18 (800 × 800)	86.7	75.4	80.7
CSFF-ResNeDt-18 (800 × 800)	87.4	77.3	82.1
DB-ResNet-50 (800 × 800)	84.3	78.4	81.3
CSFF-ResNeDt-50 (800 × 800)	86.6	78.9	82.6

Table 4. Test results on ICDAR 2015 data set. The values in parentheses represent the height of the input image. "PSE" is PSENet.

Method	P (%)	R (%)	F (%)
EAST (ZHOU et al., 2017)	83.6	73.5	78.2
Corner (Lyu et al., 2018b)	94.1	70.7	80.7
RRD (Liao et al., 2018)	85.6	79.0	82.2
PSE-1s (Wang et al., 2019a)	86.9	84.5	85.7
SPCNet (Xie et al., 2019a)	88.7	85.8	87.2
LOMO (Zhang et al., 2019)	91.3	83.5	87.2
CRAFT (Baek et al., 2019)	89.8	84.3	86.9
SAE (Tian et al., 2019)	88.3	85.0	86.6
DB-ResNet-18 (1280 × 736)	89.3	73.8	80.8
CSFF-ResNeDt-18 (1280 × 736)	86.4	79.2	82.7
DB-ResNet-50 (1280 × 736)	88.6	77.8	82.9
CSFF-ResNeDt-50 (1280 × 736)	90.6	77.3	83.4
DB-ResNet-50 (2048 × 1152)	89.8	79.3	84.2
CSFF-ResNeDt-50 (2048 × 1152)	89.6	81.1	85.1

Table 5. Test results of the algorithm on MSRA-TD500 dataset. The value in parentheses is the height of the input image.

Method	P (%)	R (%)	F (%)
(He et al., 2016b)	71.0	61.0	69.0
DeepReg (He et al., 2017b)	77.0	70.0	74.0
RRPN (Ma et al., 2018)	82.0	68.0	74.0
RRD (Liao et al., 2018)	87.0	73.0	79.0
MCN (Liu et al., 2018)	88.0	79.0	83.0
PixelLink (Deng et al., 2018)	83.0	73.2	77.8
Corner (Lyu et al., 2018b)	87.6	76.2	81.5
TextSnake (Long et al., 2018)	83.2	73.9	78.3
(Xue, Lu, and Zhan 2018)	83.0	77.4	80.1
(Xue, Lu, and Zhang 2019)	87.4	76.7	81.7
CRAFT (Baek et al., 2019)	88.2	78.2	82.9
SAE (Tian et al., 2019)	84.2	81.7	82.9
DB-ResNet-18 (512 × 512)	85.7	73.2	79.0
CSFF-ResNeDt-18 (512 × 512)	88.8	77.7	82.9
DB-ResNet-18 (736 × 736)	90.4	76.3	82.8
CSFF-ResNeDt-18 (736 × 736)	87.8	81.8	84.7
DB-ResNet-50 (736 × 736)	91.5	79.2	84.9
CSFF-ResNeDt-50 (736 × 736)	89.4	82.3	85.7

4.5.1. Curved Text Detection

In this paper, the model is also evaluated on the Total-Text dataset, which is used to demonstrate the ability of detecting curved text. Set the height of the input image to 800 according to [3,4]. As shown in Table 3, the performance of our method is greatly improved compared to that of the original network. Specifically, "CSFF-ResNeDt-18 (ResNeDt-18+Cross-Level FPN+DB)" outperforms the previous baseline method by 1.4%. Compared to previous best method TextSnake, "CSFF-ResNeDt-50 (ResNeDt-50+Cross-Level FPN+DB)" shows advantages in accuracy and F-measure, and the effect is improved by 3.9% and 4.2% respectively. The visualization results are shown in Figure 8. Experiments show that this method can effectively deal with irregular shapes and curved texts in any direction, and shows strong robustness in detecting arbitrarily bent text instances. Compared with the baseline, the results of our method have higher accuracy and can obtain more accurate boundary boxes. It is worth noting that the CSFF-ResNeDt represents the different backbones used in our network.

Both the CRAFT and the CSFF-Net proposed in this paper are segmentation-based text detection. The difference is that CRAFT mainly detects a single character and the links among characters, and then determines the final text box based on the links among characters. while the CSFF-Net generates the text box by directly detecting the text. Since character-level image segmentation is more time-consuming and introduces less background information than text line image segmentation, a better performance may be achieved. LOMO detects the text region by regression, and then obtains the final text box by using the text box center line and text box boundary offset. The CSFF-Net in this paper obtains text regions directly by segmentation. The former network model is more complex possibly learning more feature information while resulting in a longer training time, which is not hardware friendly.

4.5.2. Multi-Oriented Text Detection

ICDAR 2015 is a multi-directional English text dataset, which contains a large number of small and low-resolution text examples. For ICDAR 2015, we evaluated our model using an image height of 736 or 1152 to test its performance in multi-oriented text. In Table 4, we can see that "CSFF-ResNeDt-50 (2048 × 1152)" achieves 89.6%,81.1% and 85.1% in the precision, recall, F-measure. Generally speaking, the model exceeds the baseline by 1.9% in terms of F-measure. Compared with other advanced methods, although the

F-measure of this model on ICDAR 2015 dataset is not superior to other methods, it can also be compared with other methods. Compared with EAST, the method in this paper "CSFF-ResNeDt-50 (1280 × 736)" has improved by 7%, 7.6% and 6.9% in P, R and F respectively. For "CSFF-ResNeDt-18 (1280 × 736)", when ResNet-18 is used, the F-measure of the model reaches 82.7%.

4.5.3. Multi-Language Text Detection

The algorithm is robust for multilingual text detection. For the MSRA-TD500 dataset, the text contained in it is long and large, so large input cannot improve the performance. Therefore, we simply adjusted the height of the test images to 512 or 736 to fit our model. Table 5 shows the comparison results between this method and other advanced methods. The algorithm has high precision, recall, and F-measure, which is an advantage over most other existing algorithms on the MSRA-TD500 dataset. In this paper, "CSFF-ResNeDt-50 (ResNeDt-50+Cross-Level FPN+DB)" is superior to previous methods in terms of accuracy. For the accuracy, "CSFF-ResNeDt-50" exceeds the previous advanced method by 2.8%. With a lightweight backbone, "CSFF-ResNeDt-18 (512 × 512)" achieves a comparative accuracy compared to the most classic algorithm (Liu et al., 2018) (82.9 vs. 83.0). This proves that our model is robust for multilingual detection and can actually be used in complex natural scenarios. To summarize, this framework performs better than other existing methods in performing scene text detection tasks, and has a superior performance, and can effectively and accurately detect texts.

In the multilingual dataset, MSRA-TD500, the feature information of various texts is quite different. For example, the proportion of English text shapes is relatively small, and the white space between texts is large. Chinese text shapes are complex, and the overall proportion is relatively large. One of the advantages of CSFF-Net in this paper is that it is designed for multi-scale changes of texts, so it is sensitive to the feature information of multilingual texts and can detect texts well.

5. Discussion

Aiming at the problem of insufficient feature information extraction in complex background image classification, this paper proposes a structure for detecting arbitrary shape text in a complex background environment and successfully detects arbitrary shape text examples. The proposed Cross-Level Feature Pyramid Network (Cross-Level FPN) plays a crucial role in the training process and is used for effective feature reuse and fusion of multi-scale contextual information. Model detection accuracy is improved by using a deep weighted convolution module and a 3D-Attention module for the backbone feature extraction network to highlight the representation of important information. The efficiency and universality of our approach have been demonstrated in publicly available scene text datasets, including long, curved, oriented, and multilingual text cases. The experiments showed the superior performance of this method and have a comparable performance compared to more advanced methods. As we deepen the depth of the network to some extent and increase the multi-scale calculation, resulting in a slight increase in time during training, but it has little impact on the detection efficiency and can achieve a real-time detection effect, and the network should be further optimized in the subsequent work.

For the next step in the future, we hope that the end-to-end recognition model can be used to train this model, and we can see if its performance, robustness, and generalization ability can be transformed into a better scene text recognition system so that it can be further applied to a wider natural environment.

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

Funding: This work is supported by Natural Science Foundation of Xinjiang (No.2020D01C045), National Science Foundation of China (NSFC) under Grant No. 62166043 and Youth Fund for scientific research program of Autonomous Region (XJEDU2019Y007).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. Our datasets can be obtained from [https://www.gitmemory.com/cs-chan/Total-Text-Dataset], [https://rrc.cvc.uab.es/?ch=4&com=tasks] (12 December 2021), [http://www.iapr-tc11.org/mediawiki/index.php/MSRA_Text_Detection_500_Database_%28MSRA-TD500%29] (12 December 2021), respectively.

Conflicts of Interest: We declare no conflict of interest.

References

1. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Luo, Z. R2cnn: Rotational region cnn for orientation robust scene text detection. *arXiv* **2017**, arXiv:1706.09579.
2. He, T.; Tian, Z.; Huang, W.; Shen, C.; Qiao, Y.; Sun, C. An end-to-end textspotter with explicit alignment and attention. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5020–5029.
3. Liao, M.; Shi, B.; Bai, X.; Wang, X.; Liu, W. Textboxes: A fast text detector with a single deep neural network. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4161–4167.
4. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. Multimed.* **2018**, *2*, 3111–3122. [CrossRef]
5. Shi, B.; Bai, X.; Belongie, S. Detecting oriented text in natural images by linking segments. In Proceedings of the 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2550–2558.
6. Tian, Z.; Huang, W.; He, T.; He, P.; Qiao, Y. Detecting text in natural image with connectionist text proposal network. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 56–72.
7. Liao, M.; Zhu, Z.; Shi, B.; Xia, G.S.; 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.
8. 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.
9. 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 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5551–5560.
10. He, W.; Zhang, X.Y.; Yin, F.; Liu, C.L. Deep direct regression for multi-oriented scene text detection. In Proceedings of the 2017 IEEE/CVF International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 745–753.
11. Xue, C.; Lu, S.J.; Zhan, F.N. Accurate Scene Text Detection Through Border Semantics Awareness and Bootstrapping. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 370–387.
12. Zhang, C.; Liang, B.; Huang, Z.; En, M.; Han, J.; Ding, E.; Ding, X. Look more than once: An accurate detector for text of arbitrary shapes. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 10552–10561.
13. Ch'ng, C.K.; Chan, C.S. Total-text: A comprehensive dataset for scene text detection and recognition. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; pp. 935–942.
14. Baek, Y.M.; Lee, B.; Han, D.; Yun, S.; Lee, H. Character Region Awareness for Text Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9365–9374.
15. Yao, C.; Bai, X.; Sang, N.; Zhou, X.; Zhou, S.; Cao, Z. Scene text detection via holistic, multi-channel prediction. *arXiv* **2016**, arXiv:1606.09002.
16. Yuliang, L.; Lianwen, J.; Shuaitao, Z.; Sheng, Z. Detecting curve text in the wild: New dataset and new solution. *arXiv* **2017**, arXiv:1712.02170.
17. Lyu, P.; Liao, M.; Yao, C.; Wu, W.; Bai, X. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 67–83.
18. Deng, D.; Liu, H.; Li, X.; Cai, D. Pixellink: Detecting scene text via instance segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 6773–6780.
19. Tian, Z.; Shu, M.; Lyu, P.; Li, R.; Zhou, C.; Shen, X.; Jia, J. Learning shape-aware embedding for scene text detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4234–4243.

20. Wang, W.H.; Xie, E.; Song, X.G.; Zang, Y.H.; Wang, W.J.; Lu, T.; Yu, G.; Shen, C. Efficient and Accurate Arbitrary-Shaped Text Detection with Pixel Aggregation Network. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 8439–8448.
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
22. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
23. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
24. 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), Nancy, France, 23–26 August 2015; pp. 1156–1160.
25. Yao, C.; Bai, X.; Liu, W.Y.; Ma, Y.; Tu, Z.W. Detecting texts of arbitrary orientations in natural images. In Proceedings of the 2012 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1083–1090.
26. Epshtein, B.; Ofek, E.; Wexler, Y. Detecting text in natural scenes with stroke width transform. In Proceedings of the 2010 IEEE/CVF Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 15–17 June 2010; pp. 2963–2970.
27. Matas, J.; Chum, O.; Urban, M.; Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **2004**, *22*, 761–767. [CrossRef]
28. Zhang, Z.; Shen, W.; Yao, C.; Bai, X. Symmetry-based text line detection in natural scenes. In Proceedings of the 2015 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 2558–2567.
29. Wang, X.B.; Jiang, Y.Y.; Luo, Z.B.; Liu, C.L.; Choi, H.; Kim, J. Arbitrary Shape Scene Text Detection with Adaptive Text Region Representation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 6449–6458.
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 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 19–35.
31. Wang, Q.H.; Xie, E.; Li, X.; Hou, W.B.; 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, Long Beach, CA, USA, 16–20 June 2019; pp. 9336–9345.
32. Zhu, X.A.; Cheng, D.Z.; Zhang, Z.; Lin, S.; Dai, J.F. An Empirical Study of Spatial Attention Mechanisms in Deep Networks. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6687–6696.
33. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv* **2014**, arXiv:1312.6229.
34. Liao, M.; Wan, Z.; Yao, C.; Chen, K.; Bai, X. Real-time scene text detection with differentiable binarization. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton New York Midtown, New York, NY, USA, 7–12 February 2020; pp. 11474–11481.
35. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26–30 June 2016; pp. 770–778.
36. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
37. Gupta, A.; Vedaldi, A.; Zisserman, A. Synthetic Data for Text Localisation in Natural Images. In Proceedings of the 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26–30 June 2016; pp. 2315–2324.
38. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
39. Freeman, H.; Shapira, R. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Commun. ACM* **1975**, *18*, 409–413. [CrossRef]
40. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]