(HTBNet)Arbitrary Shape Scene Text Detection with Binarization of Hyperbolic Tangent and Cross Entropy

Zhao Chen *

Posted Date: 15 May 2024

doi: 10.20944/preprints202405.1040.v1

Keywords: Scene Text Detection; binarization; hyperbolic tangent; MSCA; FMCS; cross entropy

Article

(HTBNet) Arbitrary Shape Scene Text Detection with Binarization of Hyperbolic Tangent and Cross Entropy

Zhao Chen

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei,China; oliver@whu.edu.cn

Abstract: The existing segmentation-based scene text detection methods mostly need complicated post-processing, and the post-processing operation is separated from the training process, which greatly reduces the detection performance. The previous method, DBNet successfully simplified post-processing and integrated the post-processing into a segmentation network. However, the training process of the model took a long time for 1200 epochs and the sensitivity to texts of various scales was lacking, leading to some text instances being missed. Considering the above two problems, we design the text detection Network with Binarization of Hyperbolic Tangent(HTBNet). First of all, we propose Binarization of Hyperbolic Tangent (HTB), optimized along with which, the segmentation network can expedite the initial convergent speed by reducing the amount of epochs from 1200 to 600. Because features of different channels in the same scale feature map focus on the information of different regions in the image, to better represent the important features of all objects in the image, we devise the Multi-Scale Channel Attention(MSCA). Meanwhile considering that multi-scale objects in the image cannot be simultaneously detected, we propose a novel module named Fused Module with Channel and Spatial(FMCS), which can fuse the multi-scale feature maps from channel and spatial dimension. Finally we adopt cross entropy as the loss function, which measures the difference between predicted values and ground truths. The experimental results show that HTBNet compared with lightweight models has achieved competitive performance and speed on Total-Text(F-measure:86.0%, FPS:30) and MSRA-TD500 (F-measure:87.5%, FPS:30).

Keywords: scene text detection; binarization; hyperbolic tangent; MSCA; FMCS; cross entropy

1. Introduction

Recently, deep-learning-based[1-7] methods of scene text detection[8-11] have made rapid progress due to their wide range of practical applications such as license plate recognition[12], automatic driving[13], smart city[14] and password guessing[15]. Scene text detection[16, 17] can be regarded as a special kind of object detection task[18-23] in computer vision. The input of this task is a scene image containing text information, and the output is the bounding box or text area of each text instance. Compared with general object detection, text in scenes has the characteristics of arbitrary shape, multi-direction, and irregular arrangement. In the meantime, scene text is always with various fonts, different colors, and complex backgrounds. Therefore, scene text detection has always been a challenging task.

Generally speaking, scene text detection is mainly divided into three methods based on regression[24-29], component[30-34], and segmentation[35-42]. Among them, the regression-based scene text detection methods greatly draw on the method of general object detection. By presetting or generating a large number of proposal boxes of different sizes and shapes at different positions in the image, the regression-based models obtain the bounding boxes containing text instances through training parameters and finally adjust the shape and position of the bounding boxes to obtain the final text area. The regression-based scene text detection method network model has no complex post-processing reasoning speed, but its detection accuracy is generally inferior to the segmentation method.

 doi:10.20944/preprints202405.1040.v1

Component-based scene text detection methods disassemble text into individual components and then use relational inference to assemble the scattered components into characters. Component-based methods work well for long text, despite a relatively high loss in disassembly and reassembly.

Segmentation-based scene text detection methods are pixel-level classification and post-processing, which can more accurately describe scene text of arbitrary shape and irregular arrangement. However, most of the existing segmentation-based scene text detection methods[35, 38-42] require complex post-processing. On the one hand, the inference speed is slow, making it difficult to be used in industrial applications. On the other hand, the post-processing operation is independent of the training process, which greatly reduces the detection performance of the network. The differentiable binary network model[36] in the early stage successfully solved this problem. The post-processing operation was integrated into the training process, and the overall training and optimization were carried out, which greatly improved the detection performance of the network and achieved SOTA at that time. However, the convergence speed of the model is slow. At the same time, the sensitivity to features in different channels and scales is lacking, so it is easy to miss texts of different scales.

Aiming at the above-mentioned two problems, we contrive the text detection Network with Binarization of Hyperbolic Tangent(HTBNet). Specifically, we design the Binarization of Hyperbolic Tangent (HTB), which greatly improves the convergence speed. At the same time, the Multi-Scale Channel Attention(MSCA) and Fused Module with Channel and Spatial(FMCS) are proposed. For feature maps in different scales, the features of the two dimensions in channel and spatial are integrated to further improve the detection performance. Due to different channels of feature maps at the same scale focusing on different object regions, the MSCA module assigns different weights to different channels and adds to the network training process to adjust and optimize. Objects of different scales in an image are difficult to simultaneously and sufficiently detect, so FMCS combines features of different scales, which improves the sensitivity of feature scales, and can obtain both backbone and detail features. Combining the above two modules makes the detection performance better.

In summary, our main contributions are three-fold:

1. We propose the Binarization of Hyperbolic Tangent(HTB), leading the convergence speed during training from 1200 epochs to 600 epochs.
2. We design the cross-entropy loss function, which is differentiable, enabling the use of optimization algorithms such as gradient descent to minimize the loss function.
3. We contrive the Multi-Scale Channel Attention(MSCA) and the Fused Module with Channel and Spatial(FMCS), which interfold features from different scales in channel and spatial. Our method achieves outstanding results on Total-Text and MSRA-TD500 benchmarks.

2. Related Work

With the rapid development of deep learning[1, 3, 43-47], scene text detection also has made great progress both in the academic and industrial fields. Generally speaking, deep-learning-based text detection methods can be subdivided into three categories: regression-based methods, component-based methods, and segmentation-based methods.

2.1. Regression-Based Methods

Regression-based[24-29] methods usually enjoy simple post-processing algorithms (e.g. non maximum suppression[48]). However, most of them are limited to representing accurate bounding boxes for irregular shapes, such as curved shapes. For example, EAST[29] consisted of two stages. In stage one a fully convolutional network extracted text regions. In stage two Non-Maximum Suppression[48](NMS) was used to remove unsuitable text predictions. It could detect text at any orientation and its speed was very fast. But its accuracy was not particularly high. To improve accuracy, there was MSR[28], which was an evolved version of EAST. It introduced a multi-scale network and a novel shape regression technique for predicting dense text boundary points. These boundary points enabled precise localization of scene text with different orientations, shapes, and

 doi:10.20944/preprints202405.1040.v1

lengths. There are another series of methods that are based on improvements to SSD[49], such as TextBoxes[25], TextBoxes++[24], and DMPN[26]. TextBoxes[25] drew inspiration from SSD[49] (Single Shot MultiBox Detector) for text detection. It modifies the SSD architecture by replacing fully connected layers with convolutional layers and adapting the convolutional kernel size 3 x3 to 1x5 to handle text detection better, considering the different aspect ratios of text compared to general objects. Additionally, TextBoxes used a different set of default box aspect ratios compared to SSD, typically incorporating ratios like 1, 2, 3, 5, 7, and 10 to account for the wide variety of text aspect ratios. It employed a single deep neural network for both text detection and character-level regression, enabling efficient and fast text detection.TextBoxes[25] could only detect horizontal text, while TextBoxes++[24] extended this capability to detect multi-oriented text. This improvement involved the following three key changes. (1)Aspect Ratios of Default Boxes: The aspect ratios of the default boxes were modified to include 1, 2, 3, 5, 1/2, 1/3, and 1/5, enabling the detection of text with different aspect ratios. (2)Convolutional Kernel Size: The 1x5 convolutional kernel was changed to 3x5 for generating text box layers, which helped improve text detection. (3)Multi-Oriented Text Box Output: TextBoxes++ was designed to output multi-oriented text boxes, allowing it to detect text at various angles. Another method based on SSD was DMPN[26], which was designed for horizontal text detection. DMPN[26] achieved multi-oriented text detection by learning the position information of four points relative to multi-oriented anchors.ATTR[27] was based on the Faster R-CNN[46], a classical regression-based text detection method. ATTR[27] was a two-stage text detection method. The first stage, was similar to Faster R-CNN, utilizing CNN + RPN + ROI to obtain text proposals. The second stage involved refining these text proposals to make the predicted boxes more accurate. Regression-based methods aim to fit text boundaries, and overall, they are not as accurate as segmentation-based methods.

2.2. Component-Based methods

The component-based[30-34] scene text detection method involved text regions into individual components, using relationship inference to identify components belonging to the same text line. Then appropriate post-processing techniques to obtain the text regions.DRRG[34] employed a graph neural network to model the geometry and relationships of text, aiming for high-precision text detection. This method was capable of accommodating various text shapes, including horizontal, vertical, and curved text, making it highly applicable in the field of scene text detection.CRAFT[30] used a segmentation method that differed from traditional image segmentation. Unlike pixel-level classification for the entire image, CRAFT only predicted character centers. It consisted of two branches. One was focused on the probability of character centers, and the other was focused on character-to-character connection relationships. After post-processing, the text bounding boxes were obtained. There is also a series of methods, which are based on SSD[49], such as SegLink[31] and SegLink++[32]. The core of SegLink[31] was to transform text detection into the detection of two local elements: segment and link. The segment was a directional box that covered a part of the text content, while the link connected two adjacent segments, expressing whether these two segments belonged to the same text. The algorithm merged relevant segments into the final bounding box based on the representation of links, improving detection accuracy and training efficiency. SegLink++[32] built upon the original SegLink[31] by introducing two types of lines: attractive links and repulsive links. These two types of lines connected segments belonging to the same text region and kept segments from different text regions apart, respectively, achieving better detection results. Component-based methods work well for long texts, but the limitation lies in the loss associated with splitting and recombining.

2.3. Segmentation-Based methods

The segmentation-based[35-42, 50-52] scene text detection method is to perform pixel-wise binary classification for text and background, followed by complex post-processing to obtain the final text regions. SAE[39] embedded shape awareness and separated closely placed text instances. It addressed the problem of excessively long text lines by clustering the output of the three

 doi:10.20944/preprints202405.1040.v1

results.PAN[41] achieved arbitrary-shaped scene text detection through segmentation principles and offered both speed and scale invariance.PixelLink[35] introduced two pixel-wise predictions based on DNN: text/non-text prediction and link prediction.TextSnake[38] predicted the Text Center Line (TCL) and Text regions (TR), acquiring a general representation of scene text of arbitrary shapes. DBNet[36], based on segmentation, obtained the threshold map and the probability map. It proposed differentiable binarization, simplifying post-processing and achieving high-precision real-time text detection. DBNet++[37], based on DBNet[36] added the Adaptive Scale Fusion module， leading to higher precision. However, the DBNet[36] had a slow convergence speed during training for 1200 epochs, and there was a possibility for improvement in feature extraction from the backbone and neck. Therefore, we propose HTBNet to achieve faster convergence during training. Additionally, we design MSCA and FMCS to thoroughly integrate the features from different scales in channel and spatial, thereby improving the accuracy of the model.

3. The Proposed Method

3.1. Overview

The overall structure of HTBNet designed in this paper is illustrated in Figure 1, which consists of four components(Backbone, MSCA, FMCS, and HTB). The backbone extracts features from the input image, and then the neck(MSCA, FMCS) further processes and fuses these features. Finally, the detection heads(HTB) predict text regions based on the fused features, and post-processing is applied to obtain the final text regions. Four components of the proposed HTBNet are the following: (1) ResNet50[44] is adopted as the backbone to gain multi-scale feature maps. (2) MSCA fuses the multi-scale feature maps obtained from the backbone. (3) FMCS simultaneously integrates information of fused feature maps from MSCA in channel and spatial. (4) HTB inputs the feature maps obtained by FMCS into the corresponding prediction head to obtain the probability map and threshold map. Then, our designed hyperbolic tangent function computes the values in the probability map and threshold map to obtain the initial binary map. The final text regions are obtained after post-processing of the binary map. The structure details of the MSCA, FMCS, and HTB modules are explained in Section 3.2, Section 3.3, and Section 3.4, respectively.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2024 doi:10.20944/preprints202405.1040.v1

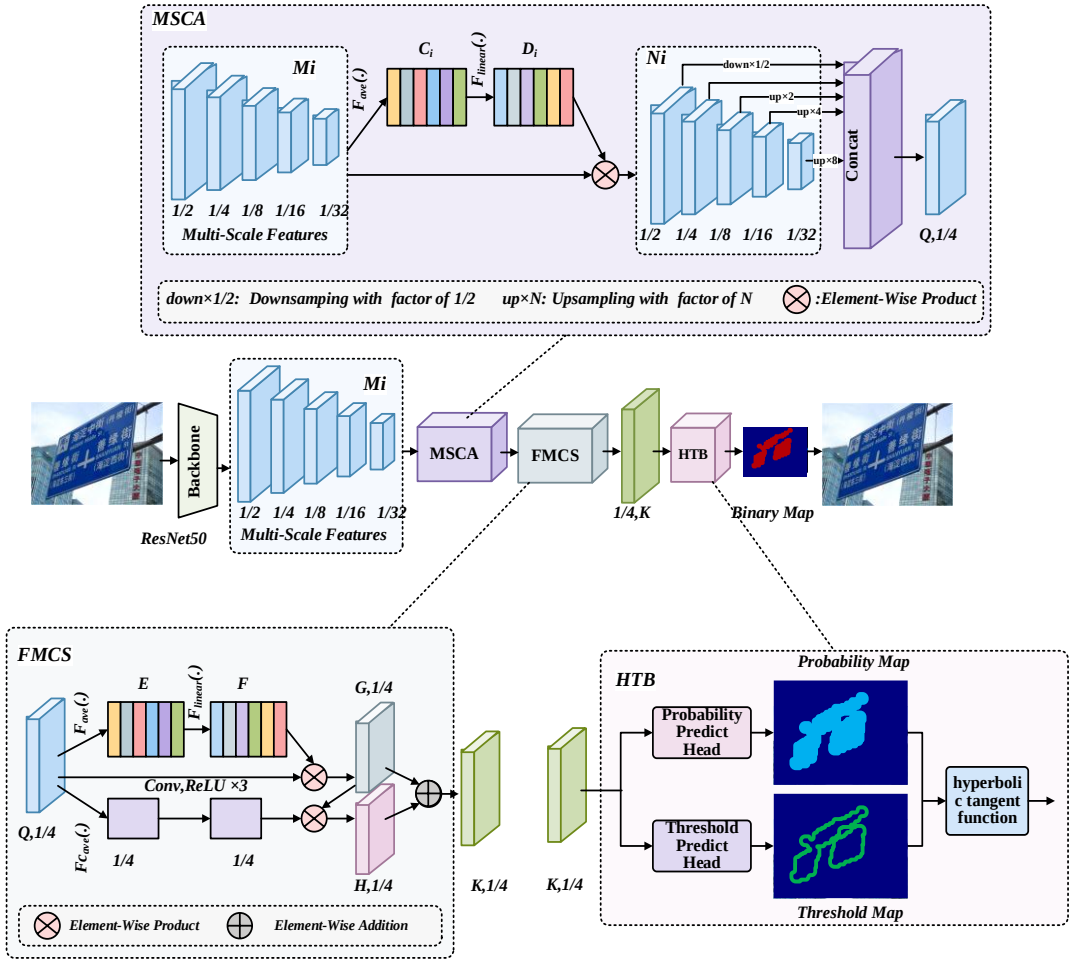

Figure 1. The Architecture of the proposed HTBNet.

3.2. Multi-Scale Channel Attention(MSCA)

The initial input of HTBNet is a scene text image, and ResNet50 serves as the backbone network to extract features, resulting in five feature maps at different scales, corresponding to 1/2, 1/4, 1/8, 1/16, and 1/32 of the original image size. Due to feature maps from different channels at the same scale focusing on different regions, we propose the novel MSCA module to better represent the kernel features of all objects in the image. The MSCA module is essentially a kind of channel attention mechanism that adds trainable weights across feature maps from different channels at the same scale. It has improved model expressiveness in the early computation of the three classical computer vision tasks (image classification, object detection and semantic segmentation). As shown in Figure 1, the input to MSCA is five feature maps at different scales, with sizes being 1/2, 1/4, 1/8, 1/16, and 1/32 of the original image. Now, the same operations are applied to each of these five feature maps. First, the feature maps at the same scale undergo global average pooling, resulting in a tensor with a dimension equal to the number of feature map's channels. Then, it passes through two fully connected layers, squeezing the dimension of this tensor to one-sixteenth, and then expanding it back to its original dimension. Finally, the new tensor multiplies channel-wise with the original feature maps. In Figure 1, 'up×N' represents upsampling the feature map N times, and 'down×1/2' represents downsampling the feature map to 1/2. Additionally, 'concat' represents concatenation operation. The above computation process can be more intuitively expressed using the following Equation 1 to Equation 3.

$$C_i = F_{ave}(M_i),(i=1,2,3,4,5) \tag{1}$$

where F_{ave} is global average pooling in the spatial dimension, and $M_i(i = 1,2,3,4,5)$ represents the feature maps at one specific scale as the input of MSCA module, meanwhile $C_i(i = 1,2,3,4,5)$

 doi:10.20944/preprints202405.1040.v1

represents a tensor with the dimension equal to the amount of corresponding $M_i's$ channels as the output of Equation 1.

$$D_i = F_{linear}(C_i) \ ,(i=1,2,3,4,5) \tag{2}$$

where F_{linear} performs two consecutive fully connected layers, and $D_i(i = 1,2,3,4,5)$ is a tensor that has the same size as $C_i(i = 1,2,3,4,5)$.

$$N_i = F_{product}(M_i, D_i) \ ,(i=1,2,3,4,5) \tag{3}$$

where $F_{product}$ refers to element-wise product, equal to M_i multiplying the $D_i's$ corresponding value of each channel by all pixel values in the spatial dimension of the corresponding channel, and $N_i(i = 1,2,3,4,5)$ has the same size as $M_i(i = 1,2,3,4,5)$. Then the multi-scale feature maps, $M_i(i = 1,2,3,4,5)$ are operated with concatenation, resulting in the new feature maps, Q, whose size is 1/4 of the original image's size.

According to the above operations of the MSCA module, subsequent ablation experiments have shown that, with little increase in the number of model parameters, significant improvement in model performance has been achieved.

3.3. Fused Module with Channel and Spatial(FMCS)

MSCA aggregates features from different channels at the same scale, but to simultaneously detect objects at different scales, we have designed FMCS to aggregate features at different scales and positions. The upper branch of FMCS merges channel information, leading to aggregating the features from all channels at different scales in the whole image. The lower branch of FMCS fuses spatial information by taking the average over the channel dimension of the feature maps, gaining a feature map with one channel. Then the feature map undergoes three convolutional operations, while kernel size is 3, and padding is 1. Next, the feature map is expanded in channel and element-wise multiplied with the feature map obtained from the upper branch. Finally, the new feature map is added element-wise to the feature map obtained from the upper branch, resulting in the final feature map of FMCS. It is important to emphasize that, Element-Wise Product and Element-Wise Addition, respectively represent element-wise multiplication and element-wise addition for two feature maps totally at the same shapes. The above process can be expressed as the following Equation 4 to Equation 5.

$$E = F_{c\_ave}(Q) \tag{4}$$

where F_{c_ave} refers to global average pooling in the channel dimension, and Q represents the feature maps as both the output of the MSCA module and the input of the FMCS module, meanwhile, as the output of Equation 4, E is a feature map with the same size as Q in spatial dimension and E has only a single channel. The functions of F_{ave}, F_{linear}, and Element-Wise Product are as same as the Equation 1, Equation 2, and Equation 3 respectively. Element-Wise Addition can be described as Equation 5.

$$K = F_{add}(G, H) \tag{5}$$

where F_{add} refers to element-wise addition, and specifically, it is that the pixel values at the corresponding positions of G and H are added element-wise to form a new feature map K. Subsequent ablation experiments indicate that the FMCS module significantly improves the model performance.

3.4. Binarization of Hyperbolic Tangent(HTB)

Based on the new feature K, obtained from the previous feature fusion, we design the HTB module to enable end-to-end training and fast convergence of the model. The probability map(P_map) and threshold map(T_map) are obtained from the corresponding prediction heads. HTB module performs exponential operations based on the difference between the values corresponding to the probability map and the threshold map using the hyperbolic tangent function. P_map and

 doi:10.20944/preprints202405.1040.v1

T_map undergo computation using our designed hyperbolic tangent function(Tanh) to obtain the initial binary map(B_map). As is shown in the following Equation 6 and Equation 7, the hyperbolic tangent function(Tanh) is an integral part of the HTB module.

$$binary\_base = \frac{e^{m} - e^{-m}}{e^{m} + e^{-m}} \tag{6}$$

$$m = k(P - T) \tag{7}$$

where P and T represent the values of the corresponding pixels on the probability map and threshold map at the same position, respectively. k is the super-parameter, and we set it to 50. The $binary\_base$ is the initial value of the feather map. If $binary\_base$ is greater than 0, then the corresponding pixel is considered to belong to the text region; otherwise, it belongs to the background region.

As seen in the subsequent experimental section, by using the hyperbolic tangent function(Tanh), the training process that initially converged in 1200 epochs is shortened to 600 epochs, significantly improving the model's convergence speed and saving on training costs and time. From a mathematical perspective, we can analyze the reason for the faster convergence of the model. As is well-known, deep learning models involve the process of backpropagation, where first-order partial derivatives are calculated for various weight coefficients. These first-order partial derivatives are then multiplied by the learning rate to obtain the corresponding weight coefficient decrement, as shown in Equation 8.

$$w' = w - lr * \frac{\partial L}{\partial w} \tag{8}$$

where L represents the total loss function, while lr represents the learning rate. And w is the initial weight coefficient, and w' is the updated weight coefficient corresponding to it. Based on the theoretical foundation above, we calculate the first-order derivatives of the two functions, and it's evident that Tanh's grad is steeper than the sigmoid's grad, which means that for the same variable step, the function value of the hyperbolic tangent function changes more significantly than sigmoid. Therefore, we can conclude that the weight coefficients of the hyperbolic tangent function decay faster than those of the sigmoid, resulting in faster convergence of the overall model.

Finally, post-processing is applied to the initial text regions obtained by HTB, involving expansion and contraction, to obtain the final text regions. It's worth noting that during the inference phase, initial text regions can be obtained using only the probability maps or threshold maps, without the necessity to compute the hyperbolic tangent function.

3.5. Cross-Entropy Loss Function

Based on the earlier computation of the probability map, the binary map, and the threshold map, by comparing the predicted values with the ground truth, we can obtain the corresponding loss functions respectively. The predictions of the probability map and the threshold map in this article are typical classification problems. The cross-entropy loss function can help the model better learn the relationships between categories. By minimizing the difference between the predicted probability distribution and the ground truths, it enhances the model's ability to classify different categories. Therefore, we use the cross-entropy loss function for both probability map loss (L_s), the binary map loss (L_b). The loss function L can be expressed as a weighted sum of the probability map loss (L_s), the binary map loss (L_b), and the threshold map loss (L_t), which refers to the specific Equation 9 to Equation 11.

$$L = L_s + L_b + 10 \times L_t \tag{9}$$

$$L_s = L_b = \sum_{i \in M} y_i \, log \, x_i + (1 - y_i) log \, (1 - x_i) \tag{10}$$

 doi:10.20944/preprints202405.1040.v1

where M represents a set where positive samples and negative samples are in the ratio of 1:3 and x_i, y_i represent the ground truth and prediction value of probability map or binary map, respectively.

$$L_t = \sum_{i \in N} |y_i^* - x_i^*| \tag{11}$$

where N represents a set of pixels within the text bounding boxes and x_i^*, y_i^* represent the ground truth and prediction value of the threshold map, respectively. With the loss functions, the entire network can undergo backpropagation and gradient computation, enabling the optimization of parameters.

4. Experiments and Results Analysis

4.1. Datasets and Evaluation

This paper focuses on natural scene text images of arbitrary shapes, so we utilize the Total-Text[53] and MSRA-TD500[54] datasets for experimental purposes. The examples of the two datasets are shown in Figures 2 and 3. Additionally, before formal training, we conduct pre-training on the SynthText [55] synthetic dataset, which is shown in Figure 4. The datasets involved in this paper are described as follows.

Total-Text (Curved Text Dataset): This dataset primarily consists of English text with a smaller portion of Chinese text. It includes 1255 images in the training set and 300 images in the test set. The text in this dataset is often curved, and it is annotated at the word level using polygons.

Figure 2. Examples of Total-Text.

Figure 3. Examples of MSRA-TD500.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2024 doi:10.20944/preprints202405.1040.v1

Figure 4. Examples of SynthText[55].

MSRA-TD500 (Multi-Oriented Scene Text Dataset): This dataset is focused on multi-oriented text detection and includes both Chinese and English text. It comprises 300 images in the training set and 200 images in the test set, with text annotated at the text-line level.

SynthText, which consists of 800k images, is a synthetic dataset used for training and evaluating text detection and recognition models. It comprises computer-generated text placed on a variety of backgrounds to simulate real-world text scenarios. This dataset offers a wide range of text appearances, including different fonts, sizes, orientations, and background textures. We utilize SynthText to pre-train our model, enhancing its ability to detect text in diverse, real-world environments.

Scene text detection is a crucial task in the field of computer vision, aiming to accurately identify and locate text regions within images captured in natural scenes. To assess the performance of text detection algorithms, quantitative analysis is often conducted using metrics such as Precision, Recall, and F1 Score. Precision is the proportion of correctly identified positive samples out of all samples predicted as positive by the model, which is calculated using the following Equation 12.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

where TP (True Positives) represents the number of samples correctly identified as positive, and FP (False Positives) represents the number of samples incorrectly identified as positive. Precision measures the accuracy of the model's positive predictions, with higher values indicating better precision in positive predictions.

Recall is the proportion of correctly identified positive samples out of all actual positive samples, which is calculated using the following Equation 13.

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

 doi:10.20944/preprints202405.1040.v1

where TP represents the number of samples correctly identified as positive, and FN (False Negatives) represents the number of actual positive samples incorrectly identified as negative. Recall measures the model's ability to identify actual positive samples, with higher values indicating broader coverage of actual positives.

F1 Score is the harmonic mean of Precision and Recall, providing a balanced assessment of a model's precision and recall. It is calculated using the Equation 14. The F1 Score ranges between 0 and 1, with higher values indicating a better balance between precision and recall. These three evaluation metrics play a crucial role in natural scene text detection. Precision focuses on the accuracy of positive predictions, recall assesses the coverage of actual positive samples, and the F1 Score combines both aspects to offer a comprehensive evaluation of model performance. In practical applications, the choice of evaluation metrics depends on task requirements, and sometimes a balance between precision and recall needs to be considered. The comprehensive consideration based on precision and recall leads to the use of the common F1 Score as the primary evaluation metric in this paper. On the other hand, there is another metric to measure the computational efficiency of the model, which is Frames Per Second (FPS). FPS measures the speed of the algorithm in processing natural scene images, indicating the number of frames that the deep learning algorithm can handle per second. A high FPS indicates that the algorithm has high real-time performance, making it suitable for practical application scenarios.

$$F1\ Score\ = \frac{2Precision * Recall}{Precision + Recall} \tag{14}$$

The experiments in this paper use the Ubuntu 20.04 operating system and PyTorch 1.12.0 deep learning framework. The training platform utilizes an NVIDIA GeForce RTX 3090 Ti graphics card with 24GB of VRAM. We first pre-train the model with the SynthText dataset for 100k iterations. Then we finetune the models on the corresponding real-world datasets for 600 epochs. It is known that there are 1200 epochs in the original baseline. The decay strategy of the learning rate, that we adopt in this paper, is SGD. The learning rate is set as the following Equation 15.

$$lr = lr * (1 - \frac{iter}{max\ \_iter})^{0.9} \tag{15}$$

where lr is the learning rate, the initial value of which is set to 0.007, and $iter$ represents the current iteration times, meanwhile $max\ \_iter$ represents the maximum iteration times.

4.2 Ablation Study

We conduct a series of ablation experiments on two datasets（i.e. Total-Text and MSRA-TD500 ） for each of the three proposed modules. The results of the ablation experiments for datasets Total-Text and MSRA-TD500 are presented in Tables 1–4.

Table 1. Detection ablation results on Total-Text with backbone of ResNet18. The 'P', 'R', and 'F' represent separately Precision, Recall, and F-measure.

Module	HTB	MSCA	FMCS	Total-Text		
				P	R	F
DB_res18(baseline)	×	×	×	88.3	77.9	82.8
res18	✓	×	×	**90.9**	77.1	83.5
res18	×	✓	×	89	78.9	83.7
res18	×	×	✓	88.5	79	83.5
HTBNet_res18(Ours)	✓	✓	✓	86.8	**81.6**	**84.1**

Table 2. Detection ablation results on Total-Text with backbone of ResNet50.

Module	HTB	MSCA	FMCS	Total-Text		
				P	R	F

 doi:10.20944/preprints202405.1040.v1

DB_res50(baseline)	×	×	×	87.1	82.5	84.7
res50	✓	×	×	**94.9**	76.8	84.9
res50	×	✓	×	87.9	82.8	85.3
res50	×	×	✓	90.5	**81.3**	**86**
HTBNet_res50(Ours)	✓	✓	✓	91.3	**81.3**	**86**

Table 3. Detection ablation results on MSRA-TD500 dataset with backbone of ResNet18.

Module	HTB	MSCA	FMCS	MSRA-TD500		
				P	R	F
DB_res18(baseline)	×	×	×	90.4	76.3	82.8
res18	✓	×	×	89.3	77.7	83.1
res18	×	✓	×	**92.3**	75.9	83.3
res18	×	×	✓	88.8	**82**	85.3
HTBNet_res18(Ours)	✓	✓	✓	89.8	81.4	**85.4**

Table 4. Detection ablation results on MSRA-TD500 dataset with backbone of ResNet50.

Module	HTB	MSCA	FMCS	MSRA-TD500		
				P	R	F
DB_res50(baseline)	×	×	×	91.5	79.2	84.9
res50	✓	×	×	90.3	81.4	85.6
res50	×	✓	×	89.7	82.3	85.8
res50	×	×	✓	91.9	**83.3**	87.4
HTBNet_res50(Ours)	✓	✓	✓	**92.2**	**83.3**	**87.5**

On the one hand, HTB accelerates the convergence speed of the model by speeding up the gradient descent. On the other hand, as can be seen from Tables 1–4, HTB significantly enhances performance. HTB has resulted in an improvement of 0.7% and 0.2% of F-measure for Total-Text dataset when using Res18 and Res50 as the backbone, respectively. And HTB module increases by 0.3% and 0.7% of F-measure for MSRA-TD500 dataset. What's more important is that HTB reduces the training process from the initial 1200 epochs to 600 epochs, greatly shortening the training time.

MSCA module, which is used to fuse the features in channel dimension, leads to more accurate features. MSCA has exhibited a significant improvement on the two datasets and the two backbone networks, with F-measure increasing by a minimum of 0.5% and a maximum of 0.9%.

FMCS aims to fuse features in both spatial and channel dimensions simultaneously. According to above Tables 1–4, FMCS achieves the highest improvement in F-measure. FMCS results in an improvement of 0.7% and 1.3% of F-measure for Total-Text dataset while Res18 and Res50 are adopted as the backbone. And FMCS increases by 2.5% of F-measure for MSRA-TD500 dataset whether the backbone is Res18 or Res50.

The model, incorporating all three modules, can get the overall results. While we use Res18 as the backbone, the F-measures improve by 1.3% on both Total-Text and MSRA-TD500. Likewise, while we use Res50 as the backbone, there is a greater improvement of 2.6% on both Total-Text and MSRA-TD500.

Randomly selecting several images from the test set for single-image testing, we can obtain visual results as shown in the following Figure 5. Especially from the two images on the left and in the middle, it can be seen that our method significantly outperforms the baseline. However, for the image on the right, due to the difficulty in distinguishing between the font and the background, both methods exhibit relatively poor detection performance.

 doi:10.20944/preprints202405.1040.v1

Figure 5. Detection results of (a)baseline and our (b)method.

4.3. Comparisons with Other Advanced Methods

We also compare our method with the previous advanced methods, and the results are presented in Tables 5 and 6. It can be observed that when we utilize Res50[44] as the backbone network, HTBNet achieves F-measures of 86% and 87.5% on datasets Total-Text and MSRA-TD500, respectively, outperforming the performance of the previously mentioned methods in Tables 5 and 6. At the same time, to make the model more lightweight, we also use Res18[44] for comparison. The model speed was significantly improved, with FPS increasing from the initial 30 to 49 and 56 on Total-Text and MSRA-TD500 respectively, and the detection performance, F1 Score remains competitive. According to Tables 5 and 6, we plot a two-dimensional scatter plot of performance versus speed, as shown in Figure 6. The horizontal axis represents the FPS of the model, and the vertical axis represents the F1 Score of the model. From both the performance and speed perspectives, our model achieved the best results.

Table 5. Detection results on Total-Text dataset. "P", "R", and "F" indicate precision, recall, and F1 Score respectively.

Methods	P	R	F	FPS
TextSnake[38]	82.7	74.5	78.4	*
PixelLink[35]	53.5	52.7	53.1	*
ATTR[27]	76.2	80.9	78.5	*
SAE[39]	82.7	77.8	80.1	*
PAN[41]	89.3	81	85	39.6
MSR[28]	73	85.2	78.6	*
DRRG[34]	84.9	**86.5**	85.7	*
DenseTextPVT[52]	89.4	80.1	84.7	*
DB++_res18[37]	87.4	79.6	83.3	48
DB++_res50[37]	88.9	83.2	86	28

 doi:10.20944/preprints202405.1040.v1

DB_res18(baseline)[36]	88.3	77.9	82.8	**50**
DB_res50(baseline)[36]	87.1	82.5	84.7	32
HTBNet_res18(Ours)	86.8	81.6	84.1	49
HTBNet_res50(Ours)	**91.3**	81.3	**86**	30

Table 6. Detection results on MSRA-TD500.

Methods	P	R	F	FPS
TextSnake[38]	83.2	73.9	78.3	1.1
PixelLink[35]	83	73.2	77.8	3
ATTR[27]	82.1	85.2	83.6	10
SAE[39]	84.2	81.7	82.9	*
PAN[41]	84.4	83.8	84.1	30.2
MSR[28]	76.7	87.4	81.7	*
DRRG[34]	82.3	**88.1**	85.1	*
PCBSNet[50]	90	76.7	82.8	*
TDGCN[51]	89.7	85.1	87.4	*
DB++_res18[37]	87.9	82.5	85.1	55
DB++_res50[37]	91.5	83.3	87.2	29
DB_res18(baseline)[36]	90.4	76.3	82.8	**62**
DB_res50(baseline)[36]	91.5	79.2	84.9	32
HTBNet_res18(Ours)	89.8	81.4	85.4	56
HTBNet_res50(Ours)	**92.2**	83.3	**87.5**	30

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 May 2024 doi:10.20944/preprints202405.1040.v1

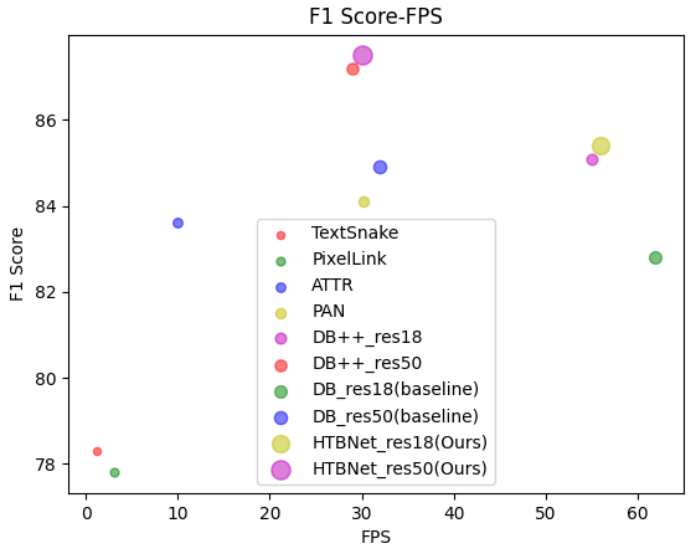

Figure 6. The two-dimensional scatter plot of performance versus speed.

In the field of text detection, looking back at lots of previous work, there has been usually focus on the model's performance, which is equal to the F1 Score, while the computational complexity of the model is frequently overlooked. However, the lightweight of deep learning models hold significant value and profound significance in practical applications in the industry. Firstly, lightweight can reduce the computational and storage resource requirements of the model, enabling more efficient deployment and operation in resource-constrained environments such as embedded devices and mobile devices. Secondly, lightweight contributes to improving the model's inference speed, reducing response time, thereby enhancing real-time capabilities, which is suitable for applications requiring rapid response, such as the scene text detection task in this paper. Additionally, lightweight can lower the energy consumption of the model, extending the device's battery life, which is crucial for battery-powered applications like mobile devices and drones. Overall, research and application of lightweight in deep learning models can propel the penetration of artificial intelligence technology into a broader range of fields, realizing more intelligent, efficient, and sustainable applications.

5. Conclusions

In this paper, we have contrived a novel framework for detecting arbitrary-shape scene text, which improves the performance of text detection from three aspects: (1) The HTB module is proposed to integrate the post-processing process into the training period and accelerate the model’s convergence during the training. (2) The designed cross-entropy loss function accurately describes the difference between the predicted values and the ground truths, which improves the model performance. (3) The proposed MSCA and FMCS extract and fuse features from channel and spatial dimensions, enhancing the model’s ability to perceive objects of different scales and positions. All of the three modules significantly improve the text detection accuracy. The experiments have verified that our HTBNet consistently outperforms outstanding methods in terms of speed and accuracy.

Author Contributions: Zhao Chen is the only author for this paper, and Zhao Chen has completed all work about this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement: The Total-text and MSRA-TD500 datasets are available at the following https://opendatalab.com/OpenDataLab/TotalText and https://opendatalab.com/OpenDataLab/MSRA-TD500, respectively.

 doi:10.20944/preprints202405.1040.v1

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

References

1. Krizhevsky, A., I. Sutskever, and G. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks.* Advances in neural information processing systems, 2012. **25**(2).
2. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* Nature, 2015. **521**(7553): p. 436-44.
3. Lecun, Y., et al., *Gradient-based learning applied to document recognition.* Proceedings of the Ieee, 1998. **86**(11): p. 2278-2324.
4. Albelwi, S. Survey on Self-Supervised Learning: Auxiliary Pretext Tasks and Contrastive Learning Methods in Imaging. Entropy, 2022. **24**, DOI: 10.3390/e24040551.
5. Lu, C. Reviewing Evolution of Learning Functions and Semantic Information Measures for Understanding Deep Learning. Entropy, 2023. **25**, DOI: 10.3390/e25050802.
6. Mazzaglia, P., et al. The Free Energy Principle for Perception and Action: A Deep Learning Perspective. Entropy, 2022. **24**, DOI: 10.3390/e24020301.
7. Vinodkumar, P.K., et al. A Survey on Deep Learning Based Segmentation, Detection and Classification for 3D Point Clouds. Entropy, 2023. **25**, DOI: 10.3390/e25040635.
8. Liu, X.Y., G.F. Meng, and C.H. Pan, *Scene text detection and recognition with advances in deep learning: a survey.* International Journal on Document Analysis and Recognition, 2019. **22**(2): p. 143-162.
9. Long, S.B., X. He, and C. Yao, *Scene Text Detection and Recognition: The Deep Learning Era.* International Journal of Computer Vision, 2021. **129**(1): p. 24.
10. Long, Y., Sun, W., Pang, Y. *et al.* Research on text detection on building surfaces in smart cities based on deep learning. *Soft Comput* **26**, 10103–10114 (2022).
11. Naiemi, F., V. Ghods, and H. Khalesi, *Scene text detection and recognition: a survey.* Multimedia Tools and Applications, 2022. **81**(14): p. 20255-20290.
12. Wang, Q., et al., *LSV-LP: Large-Scale Video-Based License Plate Detection and Recognition.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2023. **45**(1): p. 752-767.
13. Chen, T.Y., et al., WHUVID: A Large-Scale Stereo-IMU Dataset for Visual-Inertial Odometry and Autonomous Driving in Chinese Urban Scenarios. Remote Sensing, 2022. **14**(9).
14. Pan, J.P., et al., A Self-Attentive Hybrid Coding Network for 3D Change Detection in High-Resolution Optical Stereo Images. Remote Sensing, 2022. **14**(9).
15. Yu, W., et al. *A Systematic Review on Password Guessing Tasks*. Entropy, 2023. **25**, DOI: 10.3390/e25091303.
16. Gupta, N. and A.S. Jalal, Traditional to transfer learning progression on scene text detection and recognition: a survey. Artificial Intelligence Review, 2022. **55**(4): p. 3457-3502.
17. Khan, T., R. Sarkar, and A.F. Mollah, *Deep learning approaches to scene text detection: a comprehensive review.* Artificial Intelligence Review, 2021. **54**(5): p. 3239-3298.
18. Liang, T., et al., A Closer Look at the Joint Training of Object Detection and Re-Identification in Multi-Object Tracking. IEEE Transactions on Image Processing, 2023. **32**: p. 267-280.
19. Machado, E.M.S., et al. Visual Attention-Based Object Detection in Cluttered Environments. in 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). 2019.
20. Wang, Z.C., et al., AOGC: Anchor-Free Oriented Object Detection Based on Gaussian Centerness. Remote Sensing, 2023. **15**(19).
21. Wu, F.L., et al., Improved Oriented Object Detection in Remote Sensing Images Based on a Three-Point Regression Method. Remote Sensing, 2021. **13**(22).
22. Wu, Z., et al., Selecting High-Quality Proposals for Weakly Supervised Object Detection With Bottom-Up Aggregated Attention and Phase-Aware Loss. IEEE Transactions on Image Processing, 2023. **32**: p. 682-693.
23. Zhang, L.Y., et al., Constraint Loss for Rotated Object Detection in Remote Sensing Images. Remote Sensing, 2021. **13**(21).
24. Liao, M.H., B.G. Shi, and X. Bai, *TextBoxes plus plus : A Single-Shot Oriented Scene Text Detector.* Ieee Transactions on Image Processing, 2018. **27**(8): p. 3676-3690.
25. Liao, M.H., et al., *TextBoxes: A Fast Text Detector with a Single Deep Neural Network.* Thirty-First Aaai Conference on Artificial Intelligence, 2017: p. 4161-4167.
26. Liu, Y.L. and L.W. Jin, *Deep Matching Prior Network: Toward Tighter Multi-oriented Text Detection.* 30th Ieee Conference on Computer Vision and Pattern Recognition (Cvpr 2017), 2017: p. 3454-3461.
27. Wang, X.B., et al., *Arbitrary Shape Scene Text Detection with Adaptive Text Region Representation.* 2019 Ieee/Cvf Conference on Computer Vision and Pattern Recognition (Cvpr 2019), 2019: p. 6442-6451.
28. Xue, C.H., S.J. Lu, and W. Zhang, *MSR: Multi-Scale Shape Regression for Scene Text Detection.* Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019: p. 989-995.
29. Zhou, X.Y., et al., *EAST: An Efficient and Accurate Scene Text Detector.* 30th Ieee Conference on Computer Vision and Pattern Recognition (Cvpr 2017), 2017: p. 2642-2651.

30. Baek, Y., et al., *Character Region Awareness for Text Detection.* 2019 Ieee/Cvf Conference on Computer Vision and Pattern Recognition (Cvpr 2019), 2019: p. 9357-9366.
31. Shi, B.G., X. Bai, and S. Belongie, *Detecting Oriented Text in Natural Images by Linking Segments.* 30th Ieee Conference on Computer Vision and Pattern Recognition (Cvpr 2017), 2017: p. 3482-3490.
32. Tang, J., et al., SegLink plus plus : Detecting Dense and Arbitrary-shaped Scene Text by Instance-aware Component Grouping. Pattern Recognition, 2019. **96**.
33. Tian, Z., et al., *Detecting Text in Natural Image with Connectionist Text Proposal Network.* Computer Vision - Eccv 2016, Pt Viii, 2016. **9912**: p. 56-72.
34. Zhang, S.X., et al. Deep Relational Reasoning Graph Network for Arbitrary Shape Text Detection. in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020.
35. Deng, D., et al., *PixelLink: Detecting Scene Text via Instance Segmentation.* Thirty-Second Aaai Conference on Artificial Intelligence / Thirtieth Innovative Applications of Artificial Intelligence Conference / Eighth Aaai Symposium on Educational Advances in Artificial Intelligence, 2018: p. 6773-6780.
36. Liao, M.H., et al. Real-Time Scene Text Detection with Differentiable Binarization. in 34th AAAI Conference on Artificial Intelligence / 32nd Innovative Applications of Artificial Intelligence Conference / 10th AAAI Symposium on Educational Advances in Artificial Intelligence. 2020. New York, NY: Assoc Advancement Artificial Intelligence.
37. Liao, M.H., et al., *Real-Time Scene Text Detection With Differentiable Binarization and Adaptive Scale Fusion.* Ieee Transactions on Pattern Analysis and Machine Intelligence, 2023. **45**(1): p. 919-931.
38. Long, S.B., et al., *TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes.* Computer Vision - Eccv 2018, Pt Ii, 2018. **11206**: p. 19-35.
39. Tian, Z.T., et al., *Learning Shape-Aware Embedding for Scene Text Detection.* 2019 Ieee/Cvf Conference on Computer Vision and Pattern Recognition (Cvpr 2019), 2019: p. 4229-4238.
40. Wang, W.H., et al., *Shape Robust Text Detection with Progressive Scale Expansion Network.* 2019 Ieee/Cvf Conference on Computer Vision and Pattern Recognition (Cvpr 2019), 2019: p. 9328-9337.
41. Wang, W.H., et al., *Efficient and Accurate Arbitrary-Shaped Text Detection with Pixel Aggregation Network.* 2019 Ieee/Cvf International Conference on Computer Vision (Iccv 2019), 2019: p. 8439-8448.
42. Xu, Y., et al., TextField: Learning a Deep Direction Field for Irregular Scene Text Detection. IEEE Trans Image Process, 2019. **28**(11): p. 5566-5579.
43. Graves, A., A.R. Mohamed, and G. Hinton, *Speech Recognition with Deep Recurrent Neural Networks.* 2013 Ieee International Conference on Acoustics, Speech and Signal Processing (Icassp), 2013: p. 6645-6649.
44. He, K.M., et al., *Deep Residual Learning for Image Recognition.* 2016 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr), 2016: p. 770-778.
45. LeCun, Y., et al., *Backpropagation Applied to Handwritten Zip Code Recognition.* Neural Computation, 1989. **1**(4): p. 541-551.
46. Ren, S., et al., *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.* IEEE Trans Pattern Anal Mach Intell, 2017. **39**(6): p. 1137-1149.
47. Vaswani, A., et al., *Attention Is All You Need.* Advances in Neural Information Processing Systems 30 (Nips 2017), 2017. **30**.
48. Neubeck, A. and L. Van Gool, *Efficient non-maximum suppression.* 18th International Conference on Pattern Recognition, Vol 3, Proceedings, 2006: p. 850-+.
49. Liu, W., et al., *SSD: Single Shot MultiBox Detector.* Computer Vision - Eccv 2016, Pt I, 2016. **9905**: p. 21-37.
50. Lian, Z., et al. PCBSNet: A Pure Convolutional Bilateral Segmentation Network for Real-Time Natural Scene Text Detection. Electronics, 2023. **12**, DOI: 10.3390/electronics12143055.
51. Zhang, S., et al. Irregular Scene Text Detection Based on a Graph Convolutional Network. Sensors, 2023. **23**, DOI: 10.3390/s23031070.
52. Dinh, M.-T., D.-J. Choi, and G.-S. Lee DenseTextPVT: Pyramid Vision Transformer with Deep Multi-Scale Feature Refinement Network for Dense Text Detection. Sensors, 2023. **23**, DOI: 10.3390/s23135889.
53. Ch'ng, C.K. and C.S. Chan, *Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition.* 2017 14th Iapr International Conference on Document Analysis and Recognition (Icdar), Vol 1, 2017: p. 935-942.
54. Yao, C., et al., *Detecting Texts of Arbitrary Orientations in Natural Images.* 2012 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr), 2012: p. 1083-1090.
55. Gupta, A., A. Vedaldi, and A. Zisserman, *Synthetic Data for Text Localisation in Natural Images.* 2016 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr), 2016: p. 2315-2324.