Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057706

ARTICLE

Detection and Recognition of Spray Code Numbers on Can Surfaces Based on OCR

Hailong Wang[*] and Junchao Shi

School of Computer Science, Zhongyuan University of Technology, Zhengzhou, 450007, China

*Corresponding Author: Hailong Wang. Email: 5386@zut.edu.cn

Received: 26 August 2024 Accepted: 24 October 2024 Published: 03 January 2025

ABSTRACT

A two-stage algorithm based on deep learning for the detection and recognition of can bottom spray codes and numbers is proposed to address the problems of small character areas and fast production line speeds in can bottom spray code number recognition. In the coding number detection stage, Differentiable Binarization Network is used as the backbone network, combined with the Attention and Dilation Convolutions Path Aggregation Network feature fusion structure to enhance the model detection effect. In terms of text recognition, using the Scene Visual Text Recognition coding number recognition network for end-to-end training can alleviate the problem of coding recognition errors caused by image color distortion due to variations in lighting and background noise. In addition, model pruning and quantization are used to reduce the number of model parameters to meet deployment requirements in resource-constrained environments. A comparative experiment was conducted using the dataset of tank bottom spray code numbers collected on-site, and a transfer experiment was conducted using the dataset of packaging box production date. The experimental results show that the algorithm proposed in this study can effectively locate the coding of cans at different positions on the roller conveyor, and can accurately identify the coding numbers at high production line speeds. The Hmean value of the coding number detection is 97.32%, and the accuracy of the coding number recognition is 98.21%. This verifies that the algorithm proposed in this paper has high accuracy in coding number detection and recognition.

KEYWORDS

Can coding recognition; differentiable binarization network; scene visual text recognition; model pruning and quantification; transport model

1 Introduction

The printed text on the bottom of aluminum cans contains crucial information about the product's identity. This information not only assists in control and maintenance during the production process and enables future product traceability but also provides consumers with key details such as production batch numbers and dates [1]. Currently, the production dates and batch numbers of most aluminum cans are applied through spray coding on the can bottoms. Accurate recognition of these spray codes is essential for ensuring product quality, safeguarding the company's reputation, and preventing counterfeit and substandard products from entering the market. However, existing detection systems for aluminum can spray codes are inadequate, often exhibiting low detection efficiency and high costs. Additionally, factors such as high-speed production lines and varying lighting conditions in industrial environments present significant challenges for the effective detection and recognition of spray codes on can bottoms.

In industrial settings, character detection and recognition methods can generally be categorized into two types: single-character recognition and sequence character recognition. In single-character recognition, the string of characters in an image is first segmented into individual characters, which are then recognized one by one and finally recombined. Traditional single-character recognition methods often rely on manually designed features for character segmentation. For instance, Vandana et al. [2] proposed an intelligent license plate correction and extraction method based on character features, achieving adaptive detection of character contours. However, this method requires manual adjustment of constraints for license plates with large tilt angles, which significantly increases the workload during the preparation phase. The shape and size of characters can vary greatly in real-world scenarios, and traditional sliding window methods can only handle horizontal characters, failing to accurately detect and locate non-horizontal characters. On the other hand, deep learning-based single-character recognition methods can automatically learn features through neural networks, demonstrating superior performance in large datasets and complex environments. For example, Ge et al. [3] addressed the issue of motion distortion in dot-matrix characters on steel billet surfaces by proposing a multi-directional line scanning method to determine the boundaries between adjacent dot-matrix characters. They also employed a point cloud registration method to recognize characters that are prone to deformation. However, this method requires a highly controlled lighting environment, leading to instability in character matching accuracy. Veit et al. [4] utilized the Convolutional Recurrent Neural Network (CRNN) algorithm to recognize identification codes on chip surfaces. Since their approach does not involve character-level annotation and segmentation, it significantly reduces the labeling workload during the preparation phase. However, this algorithm has certain limitations when handling long character sequences and shows some shortcomings in the timeliness of character recognition. Guan et al. [5] addressed the challenge of locating bent characters on metal parts in industrial settings by proposing an optimized feature attention network, which significantly improved the quality of character candidate regions. However, this method may not fully cover the complete area of small targets within the candidate regions, potentially leading to missed detections. Niu et al. [6] replaced the Visual Geometry 16-layer network (VGG16) network in the Connectionist Text Proposal Network (CTPN) model with a Residual Network (ResNet) [7] network and incorporated an attention mechanism to capture contextual information, thereby enhancing the robustness of the detection algorithm in multilingual environments. However, this algorithm performs poorly when detecting blurred text or large-font characters.

Based on the analysis above, this paper develops a recognition network specifically tailored to the characteristics of can bottom spray codes on high-speed conveyor belts. We propose a detection and recognition algorithm for can bottom spray codes, based on an improved Differentiable Binarization Network (DBNet) [8] and the Scene Visual Text Recognition (SVTR) [9] algorithm. In the detection phase, we enhance DBNet by integrating residual attention convolutions and dilated convolutions to design an Attention and Dilation Convolutions Path Aggregation Network (AD-PAN) feature pyramid structure. This approach addresses the challenges of small and multi-scale can bottom spray code regions by fusing multi-scale features and expanding the receptive field, thereby improving the robustness of the detection process. For the recognition phase, we employ the SVTR sequence recognition network, creating an end-to-end model for training to enhance the accuracy of can bottom spray code recognition. Additionally, spray code data was collected from a production line at a canned food factory, and a corresponding dataset was constructed. Comparative experiments conducted on this dataset confirmed that the proposed algorithm achieves high recognition speed and accuracy.

2 Related Work

Can bottom spray code recognition is a form of Optical Character Recognition (OCR), specifically focused on the OCR recognition of particular character texts. In this field, several methods utilizing deep learning have already been developed and have achieved promising results.

2.1 Text Detection

Text detection is essentially a specialized form of object detection. In the field of object detection, common models include You Only Look Once (YOLO) [10], Faster Region-based Convolutional Neural Networks (Faster-RCNN) [11], and others. These models have undergone continuous optimization and improvement, becoming quite mature and capable of handling most detection tasks. However, their detection results are typically represented as horizontal rectangular boxes. For targets such as inclined text strings, these rectangles often need to include a large amount of irrelevant background to fully enclose the text, which not only increases the complexity of subsequent processing but may also negatively impact the accuracy of text recognition. Liu et al. [12] used two Bezier curves to form the top and bottom edges of the text box in Adaptive Bezier-Curve Network (ABCNet), replacing the traditional text box. A Bezier curve can represent a curve through several control points, allowing the curve to be located by predicting the coordinates of these points, thereby reducing the interference of extraneous background in the results. Yao et al. [13] approached text localization holistically, treating text detection as a semantic segmentation problem. They directly generated pixel-level prediction results from the input image and obtained text detection results through further post-processing. This method adapts well to curved and deformed text in natural scenes. Shi et al. [14] defined text as two detectable elements, segments and links, where a segment is typically a word or character. By aggregating the predicted segments based on the links, text lines can be obtained. Liao et al. [15] proposed a rotation-sensitive regression detector that extracts rotation-sensitive features through convolutional kernels in an active rotation regression branch. By pooling these rotation-sensitive features, the classification branch can extract rotation-invariant features, thereby enhancing the neural network's ability to detect text at different angles and orientations. Liao et al. [16] also proposed an end-to-end scene text detection method that does not require additional post-processing beyond non-maximum suppression, thus improving the speed of text detection.

The literature [17] introduced an innovative Efficient and Accurate Scene Text Detection (EAST) network model, which is capable of detecting quadrilateral text or text lines in any direction and shape within an image. The EAST model simplifies the detection process, completing detection in just two steps: using a fully convolutional network (FCN) and non-maximum suppression (NMS). This effectively reduces detection time and eliminates redundant intermediate processes. However, the model has lower accuracy in locating long text and multiple text instances, making it unsuitable for the requirements of this study.

In the literature [8], the authors introduced DBNet, an innovative detection framework designed to enhance the detection capabilities of targets in complex scenes by combining instance segmentation and object detection methods. DBNet employs a multi-scale feature fusion mechanism that captures target features across various resolutions and precisely locates target boundaries through efficient instance segmentation. Additionally, the authors proposed a module called Differentiable Binarization (DB), which performs binarization within the segmentation network. By optimizing the DB module, the segmentation network can adaptively set binarization thresholds, simplifying subsequent processing steps and improving text detection performance.

When text occupies a small portion of the image, a text detection step is necessary. In this experiment, using a simple rectangular bounding box to locate the text is sufficient to meet the requirements.

2.2 Text Recognition

The current Encoder-Decoder framework is the mainstream approach in the field of text recognition. In the ASTER deep learning model, its recognition module is built based on this framework. During the encoding stage, convolutional neural networks (CNN) work in tandem with bidirectional long short-term memory networks (Bi-LSTM) to extract features and capture contextual information from the text. In the decoding stage, long short-term memory networks (LSTM) and attention mechanisms further refine these features to generate the final recognition results. By utilizing recurrent neural networks (RNN) and their improved variants, the contextual relationships within the text can be effectively exploited, significantly improving text recognition accuracy [18]. However, the length of the text string can impact the model's recognition performance, as evidenced by the test results from the An Attentional Scene Text Recognizer (ASTER) network. When the string length exceeds 11 characters, accuracy tends to decline. Specifically, when the string length reaches 14 characters, the accuracy drops significantly, down to around 50%.

In recognition tasks, another approach is to segment the image containing the text into individual characters, using methods like object detection to separately locate and classify each character. This method has certain advantages when dealing with irregular text, as it does not require text rectification; instead, it directly classifies each character and combines the results into a string according to specific rules. Shi et al. [19] proposed a spatial transformation method that can convert distorted text lines into a form that is easier for subsequent networks to process, reducing the difficulty of text recognition and improving the network's robustness in handling distorted text. Baek et al. [20] demonstrated the feasibility of using only a small amount of real data for natural scene text recognition tasks. They explored the improvement of traditional training methods for natural scene text recognition through semi-supervised and self-supervised approaches, while also experimenting with various data augmentation strategies to make the most of the limited real data. Du et al. [21] used certain techniques to enhance model performance while limiting model size, proposing a very lightweight character recognition model capable of recognizing most Chinese characters, letters, and numbers. Hu et al. [22] proposed an attention-guided CTC model to learn better feature representations. This method retains the fast inference speed of the CTC model while also achieving good robustness in natural scene text recognition tasks. The convolutional character networks (CharNet) [23] network is an end-to-end text detection and recognition model that integrates both tasks. This algorithm demonstrates strong robustness in text detection and recognition within complex scenes and has achieved several state-of-the-art (SOTA) results on various datasets. CharNet first employs a 50-layer residual network from Reference [7] and two Hourglass structures [24] as the backbone to extract features from the input image. The network then splits into two major branches: one branch detects the bounding box of the text string, while the other detects and recognizes individual characters. Finally, the network combines all the detected characters within the same bounding box to form a complete string and outputs the result. Although end-to-end scene text recognition methods were implemented in studies [25], they still lack practicality due to the complex training process and slow computation speed [26]. Moreover, these methods generally rely on manually designed network architectures, and even experienced researchers often find it challenging to design a reasonable neural network structure in a short amount of time.

In summary, there are two methods for recognizing can bottom spray codes. The first method involves text localization, where the target text is extracted from the original image, and the relevant region is cropped out. The text is then recognized using a method based on recurrent convolutional networks. The second method uses object detection algorithms for character-level localization and classification, after which the characters are arranged according to their coordinates to form the recognized result string. Since can bottom spray codes typically consist of two lines of text in a single image, this paper adopts an improved localization-based recognition method.

3 The Whole Structure of Can Bottom Coding Detection and Recognition Algorithm

In this paper, we improve the DBNet model and use the enhanced algorithm to output the bounding box containing the spray code in each can bottom image. After cropping, the spray code image is input into the SVTR network, which then recognizes the characters it contains. Finally, the original image, bounding box location, and recognized character content are output simultaneously. The structure of the can bottom spray code detection and recognition algorithm is shown in Fig. 1.

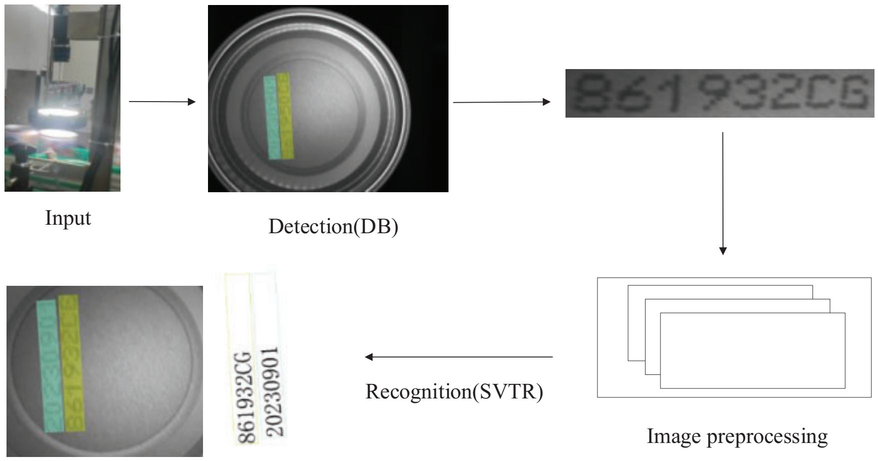


Figure 1: Overall structure of can bottom coding detection and recognition algorithm

4 Method

4.1 Improved the Detection Algorithm of DBNet Can Bottom Code

Considering that the spray code characters occupy a relatively small portion of the captured image, approximately 6%, and that the conveyor belt operates at high speed, this task falls under the category of small object detection at high speed. Therefore, this section introduces an A-PAN (attention and dilation convolutions path aggregation network) structure, designed using residual attention convolutions and dilated convolutions, to enhance multi-scale localization capabilities and expand the receptive field. The improved DBNet algorithm for can bottom spray code detection is illustrated in Fig. 2.

According to the DBNet model's processing flow, the captured can bottom spray code image is processed through the A-PAN structure to generate four feature maps, each with dimensions that are 1/4, 1/8, 1/16, and 1/32 of the input image's height and width. These feature maps are then upsampled to 1/4 of the input image's height and width, and their features are fused to produce a feature map that is 1/4 the size of the original image. This 1/4 feature map is then used to generate the Probability map and Threshold map. The Probability map and Threshold map undergo a differentiable binarization operation to produce an Approximate binary map. Finally, the loss function is computed based on the binarized map generated by the model, and model training is conducted through backpropagation.

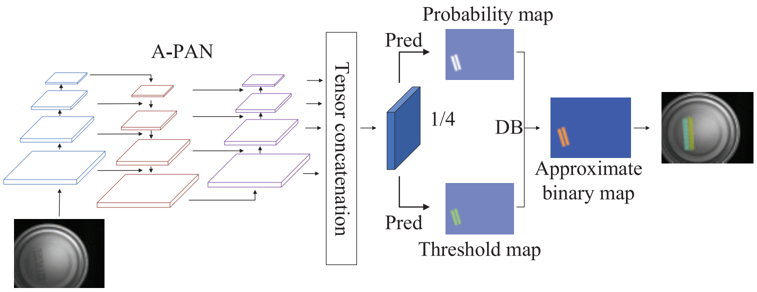


Figure 2: Improved DBNet can bottom coding detection algorithm structure diagram

As illustrated in Fig. 2 and detailed in Fig. 3, the A-PAN structure primarily serves to enhance and aggregate the image feature pathways, thereby facilitating the propagation of low-level localization information. The input image is processed by the backbone network to generate feature maps at five different scales: {C1, C2, C3, C4, C5}. To address the issue of insufficient feature representation capacity in the backbone network's output, a residual squeeze excitation (RSE) mechanism was introduced, and the convolutional layers in the model's lateral connections were replaced with residual attention convolutions. Feature map C5 is transformed into feature map P5 through residual attention convolutions and then fused with other lower-level features via a top-down propagation pathway to produce {P2, P3, P4, P5}. The newly generated feature maps corresponding to {P2, P3, P4, P5} are denoted as {N2, N3, N4, N5}, where N2 is identical to P2. Additionally, Path Aggregation Network (PANet) [27] shortens the information pathway between low-level and high-level features.

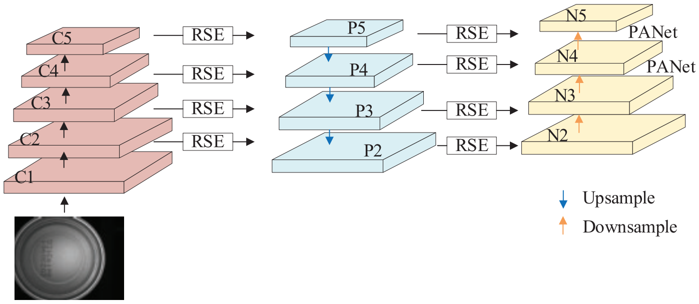


Figure 3: A-PAN structure diagram

The residual attention convolution structure in Fig. 3 is detailed in Fig. 4. In the A-PAN structure, due to the different convolution kernel sizes used during upsampling and downsampling, a 1 × 1 convolution is applied to C_i to obtain a consistent channel dimension (256). P_i is then processed with a 3 × 3 convolution, resulting in a channel dimension of 256 as well. Subsequently, the feature map passes through a sequence of layers: a 1 × 1 convolution layer, a ReLU activation layer, another 1 × 1 convolution layer, and a Sigmoid activation layer, generating corresponding spatial weights for each feature map. The resulting weight maps are multiplied with the feature maps that have been merged by channels, and the resulting feature maps are added back to the input feature maps to produce the final feature map. This final feature map, enriched with multi-scale contextual information, helps mitigate the information loss that can occur due to the reduction in the number of channels.

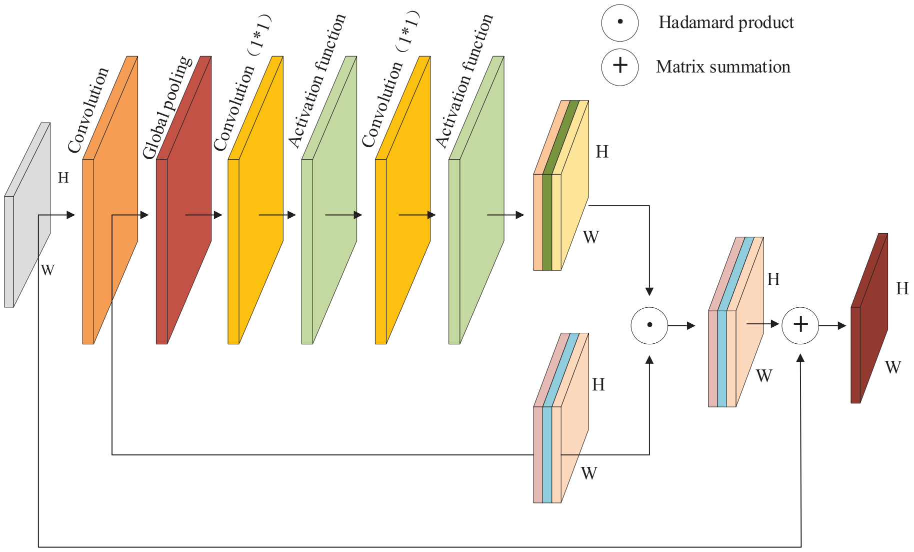


Figure 4: Residual attention convolution structure diagram

In this section, a tree-structured hierarchy, as shown in Fig. 5, is designed within the A-PAN structure to integrate multi-scale and contextual information. The tree-structured hierarchy utilizes dilated convolutions to provide different receptive fields for the input feature maps. It then employs the add operation to merge the feature information from the three parallel branches, thereby enhancing the model's multi-scale image prediction capability. In this structure, the dilated convolutions in the three parallel branches have the same convolution kernel size but different dilation rates. Specifically, each dilated convolution has a kernel size of 3 × 3, and the dilation rates d for the different branches are 1, 3, and 5, respectively. The multi-branch dilated convolution feature fusion helps the model more quickly extract the target's edge features, enhancing the clarity and accuracy of boundaries, and speeding up target localization, which is particularly critical for the detection of small targets.

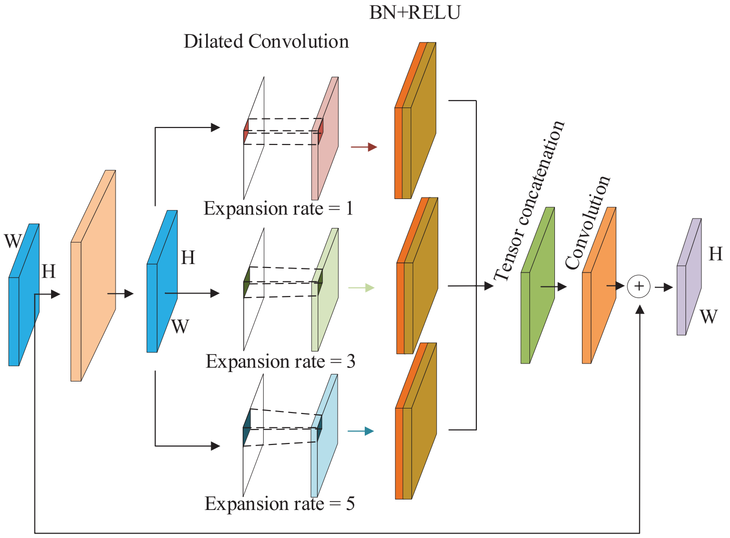
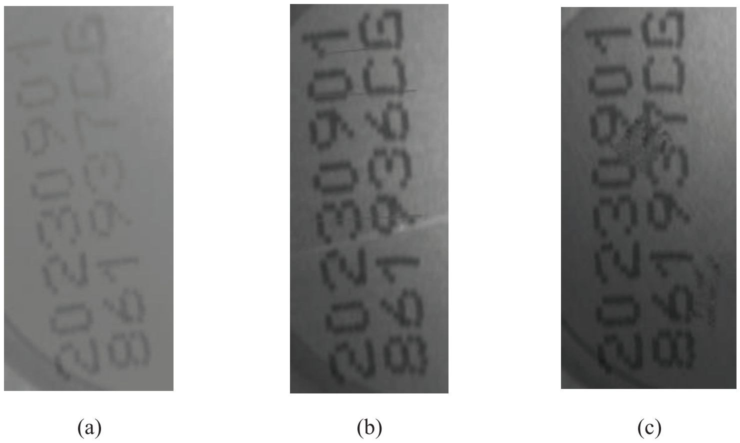


Figure 5: Hierarchical tree structure diagram

4.2 SVTR End-to-End Coding Recognition Algorithm

In the process of can bottom spray code recognition, various factors such as lighting conditions and the instability of the spray coding equipment can result in issues like low character contrast, blurred characters, or connected characters (as shown in Fig. 6). These challenges significantly complicate accurate code recognition. To address these issues, this paper introduces a SPIN correction network based on convolutional neural networks, which transforms the spray code character images in the color space. This network is integrated with the spray code recognition network to form the SPIN-SVTR end-to-end network. During model training, this integration allows the model to adaptively extract the necessary features from the input images, thereby improving the overall performance of spray code recognition. The structure of the SPIN network is shown in Fig. 7.



(a) (b) (c)

Figure 6: Inkjet characters of different types of images. (a) A blurry image taken due to changes in lighting; (b) Conjoined characters caused by the inkjet device; (c) Inkjet characters with stains

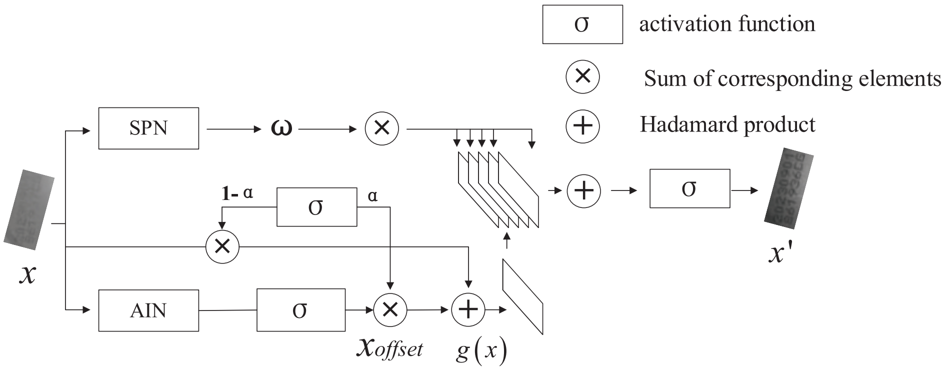


Figure 7: SPIN network structure diagram

The improved DBNet model can segment the bounding boxes of spray codes from images captured on a high-speed conveyor. Additionally, the SPIN correction network learns the mapping function of the input images to adjust the color of the spray code character images. The SPIN correction network represents all pixels with the same brightness in the spray code character image as a structural pattern. This network consists of two parts: the Structure Preserving Network (SPN) and the Auxiliary Inner-offset Network (AIN). The SPN component addresses color distortion issues caused by variations in text brightness due to lighting conditions in the spray code images. The AIN is an auxiliary network that distinguishes between color shifts caused by noise patterns such as connected characters and blurred characters resulting from spray coding equipment issues.

The Structure Preserving Network (SPN) essentially preserves the image structure by filtering the brightness levels of the input image. In the transformed image, all pixels with the same brightness level in the original image maintain the same brightness level. In this section, the structural pattern is defined as the set of pixels in the image where the brightness level is l: $\{(i,j)|x\,(i,j) = l\}$. The SPN transforms the pixel brightness of the image on a per-pixel basis into different intensity levels, thereby separating text patterns from non-text patterns while mapping the text patterns to similar brightness levels. This process aggregates different text patterns. In the SPN structure, given an input image $x \in I$, let $x' \in I'$ represent the transformed image. The transformation T from I to I' can be defined as shown in Eq. (1).

$$x'\,(i,j) = T\,[x\,(i,j)] \tag{1}$$

In the equation, $x(i,j)$ and $x'(i,j)$ represent the brightness of the input and output images at the coordinate point (i,j), respectively. The image transformation by the SPN is specifically described by Eq. (2).

$$x' = T\,(x) = sigmoid\left(\sum_i W^S_{(i)} H_{SPN}\,(x)\,x^{\beta_i}\right) \tag{2}$$

In the equation, W^S and $H_{SPN}()$ represent the partial weights and feature extractor of the final fully connected layer in the SPN, respectively. $w^s H_{SPN}()$ is the (2K+1) dimensional feature vector output by the final fully connected layer.

Since the SPN does not consider that the brightness of noise patterns might be similar to that of text patterns, it can lead to pattern confusion. However, the AIN can generate a color offset $x_{offset}(i,j)$ at each coordinate point (i,j) to mitigate this issue. Specifically, the AIN first segments the image and then calculates the offset value for each individual block. It is important to note that all offset values are activated by a sigmoid function and are mapped to the size of the input image using an upsampling operation. In the AIN, the SPN generates the color offset at each coordinate point as shown in Eq. (3).

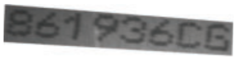
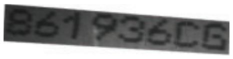
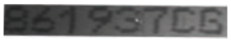
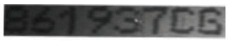
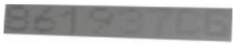
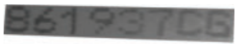
$$\begin{aligned} g\,(x) &= (1-\alpha)\otimes x + \alpha \otimes x_{offsets} \\ \alpha &= sigmoid\left(W^Z H_{SPN}\,(x)\right) \\ x_{offsets} &= W^\alpha H_{AIN}\,(x) \end{aligned} \tag{3}$$

In the equation, W^Z is the partial weight of the last fully connected layer in SPN, W^α is the weight parameter of AIN, $H_{AIN}()$ is the feature extractor, $\otimes$ is the Hadamard product. α is a learnable update gate that receives information from the SPN and senses the difficulty of different tasks. It balances the input image x with the predicted color offset. $g(x)$ is the updated image. With the assistance of AIN, the enhanced transformation of the updated image is performed as shown in Eq. (4).

$$x' = T\,(x) = sigmoid\left(\sum_i \omega_i\,(g\,(x))\,\beta^i\right) \tag{4}$$

As shown in Table 1, after SPIN preprocessing, the pixel brightness of the image characters has been effectively enhanced, while the background brightness has been suppressed. This results in a narrower grayscale range for the target area and a broader grayscale range for the background area, making the brightness contrast between the background and the target more pronounced.

Table 1: SPIN preprocessed image comparison effect

Number	Inkjet image original	After SPIN pretreatment
1		
2		
3		

In the code recognition network, the SVTR algorithm is used, which was introduced in PaddleOCRV3 [28] as a character recognition network based on transformer encoding and Connectionist TemPoral Classification (CTC) decoding. This network acquires multi-scale features through multi-stage down-sampling and processes and fuses these features using a self-attention mechanism and a multi-layer perceptron. The structure of the SVTR algorithm is shown in Fig. 8.

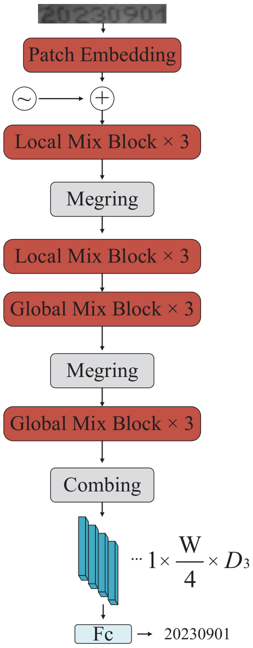


Figure 8: SVTR algorithm structure diagram

In the encoding process illustrated in Fig. 8, the spray code images corrected by the SPIN network first undergo feature extraction through a Patch Embedding layer. The feature information is then progressively transmitted and aggregated through multiple stacked Mix Blocks. Within these Mix Blocks, Merging downsampling operations are performed to reduce the height of the feature maps, thereby decreasing computational complexity. The Mix Blocks are divided into two parts: Local Mix Block and Global Mix Block. The Local Mix Block utilizes a sliding window approach for self-attention computation, which is used to capture the morphological features of the spray code characters as well as the related features between different parts of the characters. On the other hand, the Global Mix Block employs a global self-attention mechanism to assess the dependencies among all character features and to mitigate the influence of non-character features. Finally, the fully connected layer outputs the character features processed by the Mix Blocks as a feature sequence. During the subsequent CTC decoding process, consecutive duplicate labels in the feature sequence are merged into a single label, blank labels are removed, and the character sequence with the highest confidence is selected as the output.

4.3 Model Compression Method

Complex models are conducive to improving model performance, but also lead to some redundancy of parameters in the model. The method of model compression can not only solve the problem of deploying large models on resource-limited devices, but also improve model reasoning speed, reduce storage and computing costs while reducing the number of model parameters, and maintain high performance. In this study, considering that the future model will be deployed in some resource-constrained scenarios, PaddleSlim under PaddlePaddle deep learning suite is adopted for model quantification and model pruning to reduce the number of model parameters.

Model pruning improves the energy efficiency and storage of the neural network by finding the right connections, reducing the amount of storage space and computation required by the neural network by learning only the important connections without compromising accuracy. The method produces the final network by pruning unimportant connections and then retraining the remaining sparse network. The comparison of model neurons before and after pruning is shown in Fig. 9.

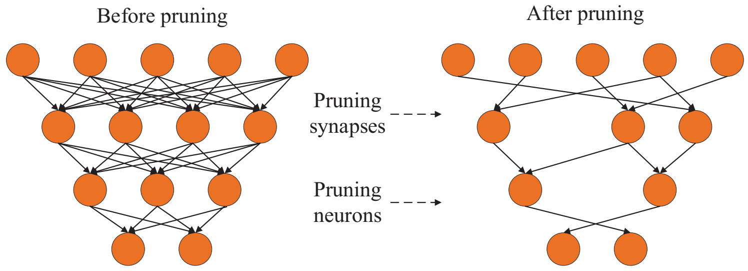


Figure 9: Comparison of neuron pruning results

Model quantization is to convert floating-point algorithm of neural network to fixed-point number to reduce the redundancy, reduce the computational complexity of the model and improve the inference performance of the model. Through model quantization, the model size can be significantly reduced with a certain loss of accuracy, which is convenient for deployment in limited resource devices. Model quantization provides online quantization and offline quantization, which can be selected according to the actual scene. The model quantization flow chart is shown in Fig. 10.

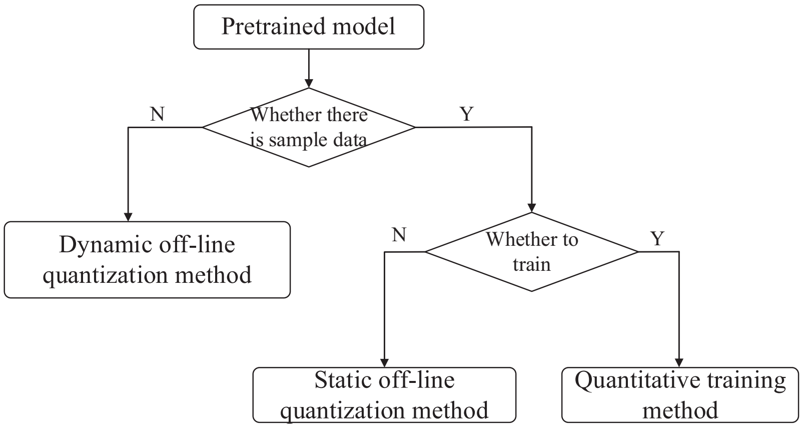

Pretrained model
Whether there is sample data
N
Y
Dynamic off-line quantization method
Whether to train
N
Y
Static off-line quantization method
Quantitative training method

Figure 10: Model quantization flow chart

5 Experimental Results and Analysis

The proposed algorithm is implemented using the PaddlePaddle-gpu 2.5.1 framework, with the programming language version being Python 3.7.16. The operating environment is the Linux Ubuntu 20.04 operating system, with a CPU model of Intel(R) Core(TM) i7-10700K running at a base frequency of 3.80 GHz. The server is equipped with 64 GB of DDR4 RAM and a Nvidia RTX A5000 graphics card with 24 GB of VRAM.

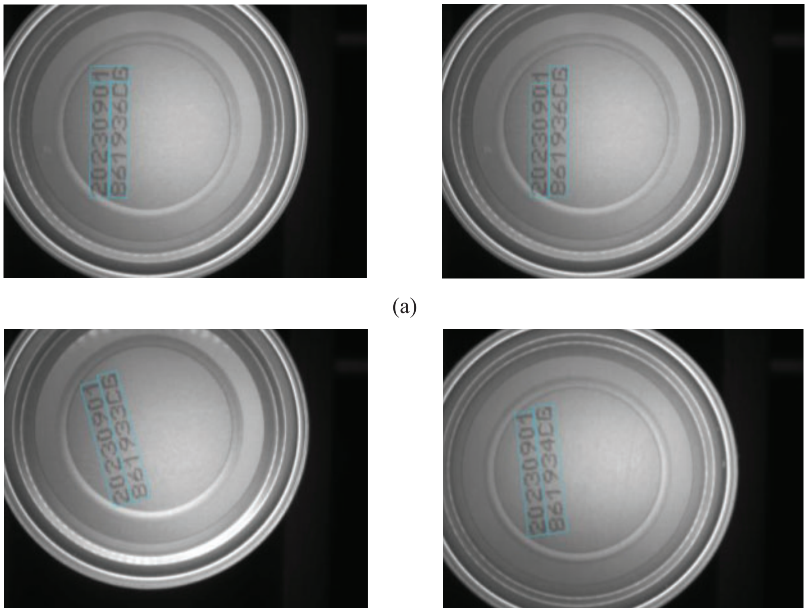
5.1 Datasets

Since the can bottom spray coding character dataset (CPS dataset) is not publicly available at present, the can bottom spray coding character dataset (CPS Dataset) selected in this paper is a real picture of the factory roller table scene. The shot data set contains characters blurred or connected characters in the bottom of the can ink-jet code due to the influence of lighting conditions, inkjet equipment and other factors. The dataset was taken over three periods, including 30 working days and a total of 10 work scenes, each of which contained 50 to 100 images. In addition, to verify the robustness of the proposed model, the complete model structure was migrated to the Box production date inkjet dataset (BPD dataset) for verification.

5.2 Improved Testing and Evaluation of DBNet Detection Algorithm

In the image code detection stage, an improved DBNet algorithm is used for network training on the code images. The input image size is uniformly adjusted to 640 × 640, with ResNet18 as the backbone network. The learning rate is set to 0.00001, and the batch size for input to the network is 8. The Adam optimizer is used, and the maximum number of iterations is set to 500. For the code image detection task, the performance of the network is evaluated using three metrics: Precision, Recall, and Hmean.

To validate the effectiveness of the improved DBNet code image detection algorithm proposed in this study, a comparative experimental analysis was conducted against the original DBNet algorithm. The experimental results are illustrated in Fig. 11, where Fig. 11a shows the detection results using the original DBNet algorithm, and Fig. 11b presents the results obtained from the improved DBNet algorithm proposed in this paper. From the comparison, it is evident that in the original DBNet detection results, the bounding boxes did not effectively enclose all the characters, leading to instances of missed detection. The root cause of this issue lies in the fact that certain character features did not respond, causing the character detection algorithm to fail to learn the features of the missed characters during training. In contrast, in Fig. 11b, the improved DBNet code detection algorithm significantly enhanced the feature representation capability for character regions, resulting in the bounding boxes accurately enclosing all characters and effectively addressing the missed detection problem observed in the original DBNet algorithm.


(a)

(b)

Figure 11: Comparison of inkjet image detection results. (a) Original DBNet algorithm detection results; (b) Improved DBNet algorithm detection results

The evaluation metrics of the improved DBNet code detection algorithm proposed in this paper, the original DBNet algorithm, and the mainstream open-source text detection algorithms EAST and PSENet [29] are compared in Table 2.

Table 2: Performance comparison between improved detection algorithm and advanced algorithm

Algorithm	Precision/%	Recall/%	Hmean/%	FPS	Params/M
EAST [17]	93.47	93.85	93.66	30.97	49.3
PSENet [29]	97.21	90.89	94.67	35.43	47.5
DBNet [8]	97.0	93.18	95.3	30.79	49.5
DBNet-L	99.13	96.72	97.6	26.32	74.2
Ours	99.01	97.1	97.32	30.5	52.7

As can be seen from Table 2, the DBNET-L inkbrush image detection algorithm improved by DBNet in this paper is superior to other detection algorithms in Table 2 in terms of Precision, Recall and Hmean indexes, when a part of FPS is lost and a part of parameter number is increased, indicating that the proposed algorithm has better robustness and accuracy. Specifically, compared with the original DBNet algorithm, the DBNET-L algorithm used in this paper has a 2.13% increase in Precision accuracy, a 3.54% increase in Recall accuracy and a 2.3% increase in Hmean accuracy.

Secondly, it is found that the DBNet-L model is too complex, and there is a negative correlation between the complexity of the model structure and the actual performance. Considering that the future model needs to be deployed in practical application scenarios, this paper adopts PaddleSlim model pruning technology to compress DBNet-L to form the final model structure. As shown in the table above, after pruning the model, the number of parameters of the model is reduced by 29%, but its performance is still better than that of DBNet-L. Compared with DBNet-L, the pruning model has a slight decrease in Precision and Hmean, while a slight increase in Recall. Although these changes have little effect on the final detection performance of the actual model, the pruned model has a higher FPS, higher detection accuracy and nearly the same FPS performance compared to other benchmark models. It was verified that PaddleSlim model pruning could reduce the complexity of the model and obtain excellent detection accuracy, which confirmed that the method adopted in this experiment could be better used in the real scene.

5.3 SVTR Coding Recognition Algorithm Test and Evaluation

During the code image recognition phase, most images in the collected dataset have an aspect ratio of 5:1. Therefore, the input image size is set to [40, 200] for this phase. A custom dictionary containing 36 characters, including digits 0–9 and letters A–Z, is used for network training. Additionally, the learning rate is set to 0.00001, the batch size is set to 128, the Adam optimizer is used, and the maximum number of iterations is set to 200.

Common evaluation metrics used in character recognition tasks include Word Accuracy (W) and Character Accuracy (C). Word Accuracy (W) represents the ratio of correctly recognized words to the total number of words, as shown in Eq. (5).

$$W = \frac{W_i}{W_{all}} \tag{5}$$

In this equation, W_i represents the total number of correctly recognized words, and W_{all} represents the total number of words to be recognized.

Character accuracy rate represents the ratio of accurately recognized characters to the total number of characters, as shown in Eq. (6).

$$C = \frac{C_i}{C_{all}} \tag{6}$$

In this equation, C_i represents the total number of correctly matched characters, and C_{all} represents the total number of characters to be recognized.

The results comparing the evaluation metrics of the improved SVTR end-to-end code recognition algorithm proposed in this paper with the original SVTR algorithm, the algorithm in Reference [4], VST [30] algorithm and the mainstream open-source text recognition algorithm ABINet [31] are shown in Table 3.

From Table 3, it can be concluded that the SVTR-L end-to-end spray code recognition algorithm, improved from the SVTR algorithm in this paper, achieves higher image recognition accuracy with minimal FPS loss and parameter addition. Specifically, the model's character accuracy improved by 2.75%, and word accuracy increased by 0.39%. Moreover, the improved SVTR code recognition algorithm outperforms other recognition algorithms in terms of parameter count and FPS. Compared to the model in Reference [4] and the ABINet algorithm, the character recognition accuracy is higher by 13.54% and 3.57%, respectively, and the code recognition accuracy is higher by 2.65% and 0.84%. The benchmark model with the best performance is the VST model, which uses a visual semantic converter to jointly model semantics and visual information to achieve the interaction between learning visual and semantic features. The proposed model in this paper improves word accuracy and character accuracy by 0.35% and 0.61% respectively compared to VST.

Table 3: Performance comparison between improved recognition algorithm and advanced algorithm

Algorithm	W/%	C/%	FPS	Params/M
Reference [4]	96.71	84.81	90.2	110.6
VST [30]	98.1	95.74	137.5	94.1
ABINet [31]	98.52	94.78	130.5	147.6
SVTR [9]	98.97	95.6	141.2	90.7
SVTR-L	99.36	98.35	136.3	99.9
Ours	99.26	98.21	140.4	93.1

In order to further control the number of model parameters, PaddleSlim's model quantization method was adopted. Through this method, the parameter count of the SVTR-L model was reduced by 6.8% compared to SVTR, while the model achieved a slight improvement in overall performance. Specifically, the quantized model showed a slight decrease in character accuracy and word accuracy compared to SVTR-L, while the recognition FPS showed a slight increase. Overall, the final model formed through model quantification can better meet the requirements while saving computational resource costs, providing a more efficient and practical solution for practical applications.

5.4 Ablation Experiment

To verify the effectiveness of the improvements introduced in the detection stage of the algorithm proposed in this paper, ablation experiments were conducted. The effectiveness of the A-PAN, RSECONV, and tree-like hierarchical structures was compared using the code image test set. The results of the ablation experiments are shown in Table 4.

Table 4: Comparison table of ablation results

A-PAN	RSECONV	Hierarchical tree structure	SPIN	Precision/%	Recall/%	Hmean/%	FPS	Params/M
√			√	96.25	93.9	95.62	30.28	50.5
√	√		√	98.76	94.31	96.16	28.59	51.1
√	√	√	√	99.13	96.72	97.6	26.32	74.2
√	√	√		97.06	93.65	95.22	26.4	69.4

From Table 4, it can be observed that when only the A-PAN structure is added to the original DBNet algorithm, the detection precision and FPS slightly decrease, but the recall rate and Hmean indicators improve, and the model's parameters increase by 1 M. With the additional incorporation of the RESCONV attention mechanism, despite a slight increase in model parameters and a decrease in FPS, there is an improvement in precision, recall rate, and Hmean indicators, with the enhanced model outperforming the original by 1.76%, 1.13%, and 0.86%, respectively. This effectively suppresses background noise and enhances the expression of target features. Further adding the tree-like hierarchical structure results in precision, recall rate, and Hmean improvements of 2.13%, 3.54%, and 2.3%, respectively. Additionally, the use of dilated convolutions with different dilation rates significantly improves the model's positive sample coverage. Although there is an increase in model parameters and a reduction in FPS, it still meets the practical requirement of FPS greater than 24 for code image generation. To verify the model's performance under significantly degraded data quality conditions, we conducted an experiment by adding other modules while removing only the SPIN image enhancement module. The results showed that although the model's performance indicators declined, there was still a slight improvement compared to the original model. This outcome not only confirmed the important role of the SPIN module in image enhancement but also provided an initial exploration of the model's behavior when faced with degraded data quality. Based on these experimental results, it is clear that the improved detection algorithm proposed in this paper effectively enhances the feature extraction capability and detection performance of code images. It also demonstrated that the model possesses a certain degree of robustness under degraded conditions. Through this experiment, the model's performance evaluation under low-quality data conditions was further refined.

6 Discussion

In the current discussion of Optical Character Recognition (OCR) technology, the development of text detection and text recognition are two core topics. With the application of deep learning technology, text detection has shifted from traditional rule-based methods to end-to-end methods based on deep learning, significantly improving the accuracy and efficiency of detection. At the same time, text recognition technology is constantly advancing. By using more complex neural network structures and large amounts of training data, modern OCR systems can better process text in various fonts, languages, and complex backgrounds.

Despite significant progress, OCR technology still faces some challenges, such as processing low-quality images, recognizing multilingual mixed text, and diverse font variations. During the research process, in order to verify the robustness of the model structure in this article, the structure was transferred to the Box production date inkjet dataset (BPDate dataset). The model detection results are shown in Table 5, and the recognition results are shown in Table 6.

In this study, the pre-trained model based on the CPS dataset was migrated to the BPD dataset and fine-tuned on the BPD dataset. As can be seen from Tables 5 and 6, the fine-tuned model still outperforms the other models on the other datasets.

In the process of model deployment, in order to ensure that the model can have a good recognition and detection ability, it is crucial to build a model training scenario in the field production environment that is as close as possible to the actual application scenario. This strategy can help improve the generalization ability and adaptability of the model, so that various situations can be predicted and processed more accurately in the actual production scenario. In order to achieve this goal, through the in-depth analysis of the production environment, the assessment of the influencing factors of the production environment, and the prediction of possible abnormal situations.

Table 5: Test model performance comparison tables in multiple data sets

Dataset	Method	Precision/%	Recall/%	Hmean/%	FPS
BPD dataset	EAST [17]	85.52	84.12	85.81	31.2
	PSENet [29]	91.61	86.25	88.37	36.13
	DBNet [8]	91.5	89.35	91.83	32.33
	DBNet-L	93.23	90.81	92.56	28.61
	Ours	93.71	91.2	92.18	31.1
CPS dataset	EAST [17]	93.47	93.85	93.66	30.97
	PSENet [29]	97.21	90.89	94.67	35.43
	DBNet [8]	97.0	93.18	95.3	30.79
	DBNet-L	99.13	96.72	97.6	26.32
	Ours	99.07	97.1	97.32	30.5

Table 6: Recognition model performance comparison tables in multiple datasets

Dataset	Method	W/%	C/%	FPS
BPD dataset	Reference [4]	81.23	83.22	88.81
	VST [30]	86.64	87.94	136.3
	ABINet [31]	88.25	89.82	134.3
	SVTR [9]	88.55	89.81	142.6
	SVTR-L	90.12	91.2	135.8
	Ours	90.3	90.64	132.7
CPS dataset	Reference [4]	96.71	84.81	90.2
	VST [30]	98.1	95.74	137.5
	ABINet [31]	98.52	94.78	130.5
	SVTR [9]	98.97	95.6	141.2
	SVTR-L	99.36	98.35	136.3
	Ours	99.26	98.21	140.5

After model deployment, continuous adjustment of model parameters and optimization algorithm using data sets can improve the accuracy and robustness of the model in different scenarios. At the same time, the model is continuously monitored and regularly updated and maintained. This includes steps such as collecting new data, retraining the model, and adjusting the model parameters. Through model iteration and optimization, it can ensure that the model always maintains good recognition and detection ability in the production environment.

7 Conclusion

This article studies the problem of inkjet image detection and recognition based on improved DBNet and SVTR algorithms. Among them, the A-PAN structure was designed based on the inkjet image detection algorithm, and residual attention convolution and dilated convolution were used to greatly improve the accuracy of inkjet image detection; Integrating the SPIN correction network into the coding image recognition algorithm for end-to-end training effectively alleviates the white balance problem caused by changes in lighting intensity and background noise in the collected coding images. By comparing the model performance on two datasets using two types of real-life collected spray code images, the experimental results show that the algorithm proposed in this paper has the following significant advantages: the compressed spray code image detection model can effectively detect the spray code at any position on the roller conveyor in the image while reducing the number of model parameters, and the Hmean of spray code detection reaches 97.32%, with a frame value of 30 frames per second, proving that the detection algorithm has good representational power; After the SPIN preprocessing method, the model effectively improved the image quality of low-quality characters such as changes in lighting conditions and connected characters in characters, proving that the model has good robustness; The recognition rate of the inkjet image detection and recognition algorithm used in this article reaches 98.21%. In the next research work, after model deployment, the model will be trained on other scenario data to gradually enhance its transferability, which can meet the needs of more scenarios for spray code detection and recognition.

Acknowledgement: I would like to thank Hailong Wang for his strong support for this study, and his careful guidance and unremitting support helped me overcome many difficulties. I would also like to express my sincere thanks to my colleagues and friends who participated in the discussion and provided valuable suggestions.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Hailong Wang; data collection: Junchao Shi; analysis and interpretation of results: Junchao Shi, Hailong Wang; draft manuscript preparation: Junchao Shi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data not available due to commercial restrictions.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] L. Belussi and N. Hirata, "Fast QR code detection in arbitrarily acquired images," in *2011 24th SIBGRAPI Conf. Graph. Patterns Images*, IEEE, 2011, pp. 281–288. doi: 10.1109/sibgrapi.2011.16.

[2] P. Vandana and B. Kaur, "A novel technique for LED dot-matrix text detection and recognition for non-uniform color system," in *2016 Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, IEEE, 2016, pp. 2750–2754. doi: 10.1109/icacci.2016.7732478.

[3] J. Ge *et al.*, "Automatic recognition of hot spray marking dot-matrix characters for steel-slab industry," *J. Intel. Manuf.*, vol. 34, no. 2, pp. 1–16, 2023. doi: 10.1007/s10845-021-01830-y.

[4] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "COCO-Text: Dataset and benchmark for text detection and recognition in natural images," 2016, *arXiv:1601.07140*.

[5] T. Guan *et al.*, "Industrial scene text detection with refined feature-attentive network," *IEEE Trans. Circ. Syst. Vid.*, vol. 32, no. 9, pp. 6073–6085, 2022. doi: 10.1109/TCSVT.2022.3156390.

[6] S. Niu, X. Li, M. Wang, and Y. Li, "A modified method for scene text detection by ResNet," *Comput. Mater. Contin.*, vol. 65, no. 3, pp. 2233–2245, 2020. doi: 10.32604/cmc.2020.09471.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778. doi: 10.1109/cvpr.2016.90.

[8] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 11474–11481, 2020. doi: 10.1609/aaai.v34i07.6812.

[9] Y. Du *et al.*, "Svtr: Scene text recognition with a single visual model," in *Europ. Conf. Artif. Intell.*, 2022, pp. 884–890. doi: 10.24963/ijcai.2022/124.

[10] J. Redmon, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016. doi: 10.1109/cvpr.2016.91.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2016. doi: 10.1109/TPAMI.2016.2577031.

[12] Y. Liu, H. Chen, C. Shen, T. He, L. Jin and L. Wang, "ABCNet: Real-time scene text spotting with adaptive bezier-curve network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9809–9818. doi: 10.1109/cvpr42600.2020.00983.

[13] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou and Z. Cao, "Scene text detection via holistic, multi-channel prediction," 2016, *arXiv:1606.09002*.

[14] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2550–2558. doi: 10.1109/cvpr.2017.371.

[15] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5909–5918. doi: 10.1109/cvpr.2018.00619.

[16] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, 2018. doi: 10.1109/TIP.2018.2825107.

[17] X. Zhou *et al.*, "East: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5551–5560. doi: 10.1109/cvpr.2017.283.

[18] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Comput. Vis.-ECCV 2016: 14th Europ. Conf.*, Amsterdam, The Netherlands, Springer, Oct. 11–14, 2016, pp. 56–72. doi: 10.1007/978-3-319-46484-8_4.

[19] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4168–4176. doi: 10.1109/cvpr.2016.452.

[20] J. Baek, Y. Matsui, and K. Aizawa, "What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3113–3122. doi: 10.1109/cvpr46437.2021.00313.

[21] Y. Du *et al.*, "PP-OCR: A practical ultra lightweight OCR system," 2020, *arXiv:2009.09941*.

[22] W. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin, "GTC: Guided training of CTC towards efficient and accurate scene text recognition," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 11005–11012, 2020. doi: 10.1609/aaai.v34i07.6735.

[23] L. Xing, Z. Tian, W. Huang, and M. R. Scott, "Convolutional character networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9126–9136. doi: 10.1109/iccv.2019.00922.

[24] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 879–886. doi: 10.1109/iccvw.2017.107.

[25] H. Li, P. Wang, and C. Shen, "Towards end-to-end text spotting with convolutional recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5238–5246. doi: 10.1109/iccv.2017.560.

[26] M. Busta, L. Neumann, and J. Matas, "Deep textSpotter: An end-to-end trainable scene text localization and recognition framework," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2204–2212. doi: 10.1109/iccv.2017.242.

[27] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768. doi: 10.1109/cvpr.2018.00913.

[28] C. Li *et al.*, "PP-OCRv3: More attempts for the improvement of ultra lightweight OCR system," 2022, *arXiv:2206.03001*.

[29] H. Nguyen, D. Tran, K. Nguyen, and R. Nguyen, "PSENet: Progressive self-enhancement network for unsupervised extreme-light image enhancement," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 1756–1765. doi: 10.1109/wacv56688.2023.00180.

[30] X. Tang, Y. Lai, Y. Liu, Y. Fu, and R. Fang, "Visual-semantic transformer for scene text recognition," 2021, *arXiv:2112.00948*.

[31] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7098–7107. doi: 10.1109/cvpr46437.2021.00702.