Data-Driven Container Marking Detection and Recognition System With an Open Large-Scale Scene Text Dataset

Ying Xu, Zhangzhao Liang, Yanyang Liang, Xinru Li, Wenfeng Pan, Jie You, Zhihao Long, Yikui Zhai, *Senior Member, IEEE*, Angelo Genovese, *Senior Member, IEEE*, Vincenzo Piuri, *Fellow, IEEE*, and Fabio Scotti, *Senior Member, IEEE*

***Abstract*—With the widespread use of containers, the demand for Container Marking Detection and Recognition (CMDR) is gradually increasing. The use of deep learning algorithms can greatly improve the efficiency of marking detection and recognition. However, there is still a lack of research on CMDR in both academia and industry, resulting in the current task being completed manually and inefficiently. In this paper, we probe into the importance of data-driven and task paradigms for CMDR tasks. Firstly, we constructed an open large scale container surface marking text dataset called ContainerText. This dataset consists of 12 k high-resolution images and provides two types of annotation information: bounding box used for detection and text for recognition tasks. In addition, we also propose an efficient semi-automatic annotation method based on deep learning, which reduces the cost of manual annotation. Subsequently, we have innovatively proposed a CMDR method combining Scene Text Recognition (STR) with CMDR tasks. The method based on STR can locate and recognize container marking from a fine-grained level. We conducted a comprehensive series of experiments on the ContainerText dataset using state-of-the-art (SOTA) scene text detection and scene text recognition models. The experimental results demonstrate that the CMDR method, based on STR, exhibits exceptional adaptability and feasibility. All experimental results obtained from the ContainerText dataset will act as performance benchmarks for future researchers. Finally, an automated Container Marking Image Acquisition Mechanism (CMIAM) are construucted, which can effectively avoid complex lighting in the workshop environment and achieve high-quality and automated image acquisition. We have conducted extensive experiments to measure the distance, resolution, and field of view required for clearly capturing container markings. Our research providing reference for future CMDR research from task solution and hardware selection.**

***Index Terms*—Container marking detection and recognition, deep learning, scene text dataset, scene text recognition.**

I. Introduction

CONTAINERS are crucial carriers in modern transportation methods. With the maturity and widespread application of deep learning technology [1], [2], [3], [4], [5], deep learning based methods have gradually gained favor from practitioners in various fields. In language processing, [6] proposed a Quantum-inspired Multimodal Sentiment Analysis (QMSA) framework, which was used to sentiment analysis. The experimental results demonstrate that the proposed framework outperforms a number of state-of-art algorithms. On the other hand, the field of computer vision has also introduced a large number of deep learning based methods for image processing. Ref. [7] introduced an integrated learning approach for low-light image enhancement. Firstly, VIT low-light enhancement sub-network provide local high-level features through a full-scale architecture, then the complementary learning sub-network is utilized to provide global fine-tuned features through learning transfer. Ref. [8] proposed a Transformer-based Generative Adversarial Network (Transformer-GAN), which used multi-head multi-covariance self-attention (MHMCA) and Light Feature-Forward Module structures (LFFM) for low-light images enhancing. The above two works effectively solve the performance degradation caused by low light shooting environment on industrial and civilian devices. In biomedicine, deep learning is also used in gene related research. Ref. [9] proposed multiscale convolution with self-attention networks (MCANet) to predict the Polyadenylation [Poly(A)] signal. This research has made significant contributions to understand the mechanism of translation regulation and mRNA metabolism. Guo et al. [10] presented a variational gated autoencoder-based feature extraction model for inferring disease-miRNA associations. In the field of containers, deep

Manuscript received 10 September 2023; revised 2 February 2024; accepted 29 February 2024. This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515011576, in part by the Guangdong Science and Technology Planning Project under Grant 2021A0505030080, Grant 2021A0505060011, Grant pdjh2022b0528, and Grant pdjh2024a374, in part by Guangdong Higher Education Innovation and Strengthening School Project under Grant 2020ZDZX3031, Grant 2022ZDZX1032, and Grant 2023ZDZX1029, in part by Wuyi University Hong Kong and Macao Joint Research and Development Fund under Grant 2022WGALH19, and in part by the Guangdong Jiangmen Science and Technology Research Project under Grant 2220002000246 and Grant 2023760300070008390. *(Ying Xu, Zhangzhao Liang, and Yanyang Liang are contributed equally to this work.) (Corresponding author: Yikui Zhai.)*

Ying Xu, Zhangzhao Liang, Yanyang Liang, Xinru Li, Wenfeng Pan, Jie You, Zhihao Long, and Yikui Zhai are with the School of Electronics and Information Engineering, Wuyi University, Jiangmen 529020, China (e-mail: xuying117@163.com; zhangzhaoliang416@163.com; liangyanyang@163.com; lxr2240743285@163.com; wenfengpan97@163.com; yau_kit@163.com; longzhihao@wyu.edu.cn; yikuizhai@163.com).

Angelo Genovese, Vincenzo Piuri, and Fabio Scotti are with the Depatimento di Information, Università Degli Studi di Milano, 20133 Milano, Italy (e-mail: angelo.genovese@unimi.it; vincenzo.piuri@unimi.it; fabio.scotti@unimi.it).

Code and ContainerText dataset can be accessed from: https://github.com/yikuizhai/ContainerText.

Recommended for acceptance by Prof. H. Cai.

Digital Object Identifier 10.1109/TETCI.2024.3377680

learning has also been applied to tasks such as container surface corrosion detection [11] and sealing detection [12], [13]. At present, the working methods of most CMDR are still not intelligent and require a lot of manual operations, which cannot meet the rapid growth of container usage and throughput. Obviously, the CMDR urgently needs to combine with artificial intelligence to reduce labor costs and improve work efficiency. What's more, the application of deep learning technology in CMDR task can overcome the limitations of manual tasks (e.g. safety issues, physical issues). Unfortunately, there is still a lack of dataset on container marking in both academia and industry at present, which hindering the application of deep learning in CMDR tasks seriously as deep learning being a data-driven technology. Ref. [14], [15] have conducted in-depth research on container code detection and recognition, and have made excellent progress. But the dataset they used are still private and lacks annotations for other markings except the container code.

To facilitate the integration of CMDR and deep learning, we have constructed an open large-scale text dataset called ContainerText, specifically designed for container markings. The dataset consists of more than 9,000 training images and more than 3,000 test images. These image inside the ContainerText dataset were captured using different devices in different scenes, with resolutions of 1920 $\times$ 1080 or higher, demonstrating rich diversity and data generalization capabilities. The annotation process of ContainerText datasets requires a great deal of time and energy. We propose a semi-automatic annotation method so as to quickly and accurately label container text. This method uses pre-trained neural networks for auxiliary annotation, combined with incremental learning and transfer learning to fully utilize the new data generated by annotation, effectively improving the speed and accuracy of annotation.

On the other hand, the numerous types of container marking, extensive customization of markings, and substantial variations in text content within these markings have rendered conventional image classification and object detection methods incapable of accurately distinguishing markings. Distinct container type codes, namely "22G1" and "45G1", exhibit high similarity in appearance features such as shape, size, and color. Despite their visual resemblance, these codes convey entirely different semantic information. Coarse-grained classification methods often erroneously categorize "22G1" and "45G1" type codes as the same class due to their similar appearances. In contrast, our method based on STR adeptly discerns the nuances between the two by recognizing finer-grained textual details. Technically, we employ the STR method to locate and recognize the text in container marking. Compared with simple classification methods, text recognition can recognize marking at a more fine-grained level, avoiding mis-classification caused by different text contents in the same marking. We conducted extensive experiments on the containertext dataset using SOTA text detection and recognition models. The experimental results demonstrate the exceptional adaptability and feasibility of the presented method.

During data acquisition, the complex workshop background and chaotic illuminants seriously affect the quality of the collected images. Therefore, we built an intelligent CMIAM on the container production line, which is composed of wide-angle cameras, telephoto cameras, infrared sensors and adjustable illuminant. The CMIAM can shield the surrounding environment that is not conducive to collecting high-quality images and achieve the intelligence and automation of CMDR tasks. With the CMIAM, we can automatically collect relevant images of container marking.

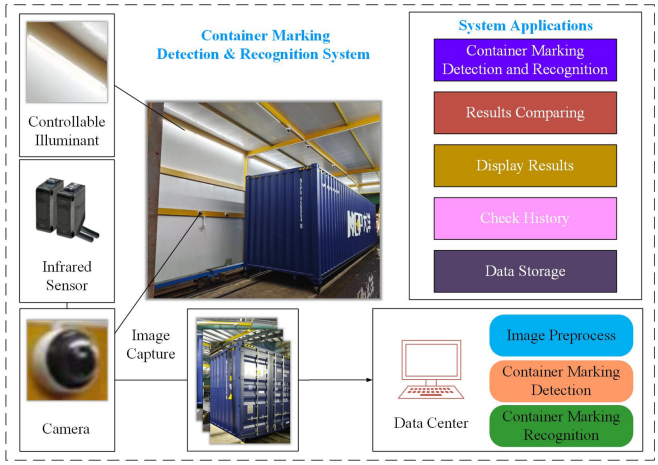


Fig. 1. Proposed CMDR system. Image acquisition component of the CMDR system comprises infrared sensors, cameras, and adjustable illuminant, collectively referred to as the CMIAM. The CMDR method, which is based on STR, is implemented in the Data Center. Once the images collected by the CMIAM are transferred to the Data Center, they undergo a series of processes including data preprocessing, marking localization, text recognition, and output the recognition results ultimately. System possesses the capability to autonomously perform CMDR tasks, provide visual representations of CMDR results, and store records.

The CMDR system is composed of the CMIAM and the STR based CMDR method. Fig. 1 shows the details of the CMDR system. All experimental data on the ContainerText dataset and key parameters of CMIAM will be made public. This research presents the following key contributions.

1) An open large-scale dataset named ContainerText is constructed for CMDR, which comprise 12,000 high-resolution images captured in real world. As far as we know, this represents the largest and the most varied publicly available dataset of container scene text. In order to efficiently and accurately complete data labeling, we proposed a semi-automatic annotation method that serves as a foundational annotation framework. It is adaptable to prevalent deep learning annotation tasks.
2) A STR based method is proposed for fine-grained CMDR, which can avoid errors caused by coarse granularity in image classification and object detection. Besides, a CMIAM is built for achieving high-quality and automatically container surface image acquisition and CMDR tasks. The proposed CMDR system consist of the STR based CMDR method and the CMIAM, which realized the intelligence of CMDR tasks and reduced the cost while improving work efficiency.
3) We verified the feasibility of the STR based CMDR method through extensive experiments using the SOTA model on the ContainerText dataset. The experimental results from the ContainerText dataset can be used as a performance benchmark.

II. Related Work

In this part, we will present commonly used scenario text data in academia and analyze the characteristics of these datasets. Secondly, we will comprehensively review the development of scene text detection and recognition techniques.

A. Scene Text Datasets

Scene text exhibits diverse text styles, background noise, text distortion, multi-scale text, and multilingualism. For document analysis competitions, Karatzas et al. introduced ICDAR2013 [16] and ICDAR2015 [17], which offer word-level annotations. Images in these datasets are natural scene and the annotation information can be available for text detection and recognition. Though these datasets widely used in STR research, it cannot be well generalized to various scenarios as theirs small scale. In order to address the high demand for data in neural network training, Gupta [18] et al. introduced a massive synthetic text dataset. The dataset utilizes natural images as background images and overlays synthesized text onto them, comprising over 0.85 million images. In the past few years, more and more challenging datasets have been proposed to solve more complex STR problems. Total-text [19] and CTW1500 [20] datasets are intended for the curve text task. These datasets consist primarily of images featuring clearly visible trademarks or signboards that contain at least one curved text. Polygon annotations are provided for these images. In addition, Singh et al. [21] have developed an arbitrary-shaped scene text dataset called TextOCR. This dataset comprises over 28 k images, where the text is oriented in multiple directions. For multi-lingual STR tasks, ICDAR2017-MLT [22] and ICDAR2019-MLT [23] are natural scene text dataset with up to 9 or 10 different scripts, including chinese, english, arabic etc. The rectangle annotation used for text localization and the word-level annotation used for text recognition.

The power of deep neural networks in feature extraction has significantly facilitated the widespread implementation of scene text detection and recognition techniques in the industry. However, the aforementioned datasets are limited to natural scene texts, highlighting a scarcity of available open-source datasets in the realm of CMDR. Some researches in the transportation industry construct container number dataset and use STR technology for container number recognition. But these data are private and small scale, which only including the annotation of container number and ignoring the rest markings that on the surface of container. To address the CMDR task and advance deep learning in container-related fields, we have introduced a container dataset of unparalleled scale called ContainerText. Table I illustrates a comprehensive comparison of the dataset scale between the ContainerText dataset and other scene text datasets.

TABLE I
ContainerText vs Others STR Datasets

Dataset	Images	Instances
ICDAR2013 [16]	462	1943
ICDAR2015 [17]	1,500	6,545
SynthText [18]	800k	5.5m
Total-text [19]	1,555	11,491
CTW1500 [20]	1,500	-
TextOCR [21]	28,134	903,069
ICDAR2017-MLT [22]	18,000	85,094
ICDAR2019-MLT [23]	20,000	89,407
ContainerText(ours)	**12,443**	**141,651**

B. Deep Learning Based Scene Text Technologies

STR is a development branch of Optical Character Recognition (OCR) task. In contrast to OCR, STR has transitioned from recognizing text in scanned electronic documents to identifying text in real world scenes. Although the scene and difficulty of the task may vary, character recognition typically involves a task flow that begins with using a text detector to identify text regions in the image on a line-by-line basis. Subsequently, the detected text lines are forwarded to the text recognizer for character recognition. Therefore, the task of STR can be further classified as two components: Scene Text Detection and Scene Text Recognition. In the past few years, deep neural networks have gained significant attention in the STR field due to their strong anti-interference capabilities and advanced feature learning.

Text Detection based on Deep Learning (DL) can be classified into two main methods: regression based and segmentation based. The regression based methods incorporates concepts from object detection, where text shapes and locations are represented as vectors through parameterization methods for regression. TextDCT [24] converts the text instance mask to the frequency domain through discrete cosine transform (DCT), extracts low-frequency components to represent the text instance mask, and finally simultaneously predicts the text instance confidence, text region coordinates, and mask vectors transformed by DCT. CT-Net [25] combines the idea of two-stage regression, first performing rough regression on the text area, then learning the offset of the text area boundary through contour refinement modules for fine regression, and finally generating the bounding box of the text area. LRANet [26] propose a parameterized text shape method. It represents text shapes by leveraging a linear combination of eigenvectors learned from arbitrary-shaped text contours. For another, the idea of segmentation based methods originates from semantic segmentation, which firstly modeling text instances with pixel-level classification masks, and then formed them into text boundaries through specific post processing. Ref. [27] used Fully Convolutional Networks (FCN) to classify text regions at the pixel level, which was an early application of segmentation ideas to text detection. Ref. [28] first segments pixels in the same instance by linking them together, and then directly extracts text bounding boxes from the segmentation results without positional regression. Ref. [29] proposed using learnable thresholds instead of fixed thresholds to binarize the segmentation maps predicted by the model, effectively improving the detection accuracy of text instances.

Text Recognition usually needs to first locate the potential text area by Text Detection, then recognize each character of the text through the word recognizer, and finally form a string by combining character sequences. In previous studies [30], [31], a common approach is to extract features from text regions using Convolutional Neural Network (CNN). Subsequently, BiLSTM was employed to capture contextual information between characters, and finally, a string sequence was decoded through prediction of the transcription stage using CTC Loss [32]. However, this method is complex and cannot effectively identify distorted characters such as deformed and blurred characters. Recently, there has been significant attention given to encoder-decoder based auto-regressive methods due to their outstanding performance. Ref. [33], [34], [35] propose to transform the recognition process into an iterative decoding process. This method greatly improves the recognition precision of distorted characters, but reduces the inference speed. To enhance the model's speed without compromising its recognition rate, researchers use complex training paradigms but use simple models for inference. Ref. [36] used attention mechanisms and graph neural networks to assemble sequence features corresponding to the same feature. When reasoning, attention to the modeling branch is ignored in order to balance performance and speed. Ref. [37] introduced a character level closed learning approach to enhance language proficiency in visual models. When reasoning, only visual models are used to improve model speed. Ref. [38] presented parallel decoding decoder that perceives context variables using dedicated designed counting and ordering modules and losses.

III. ContainerText Dataset

This part begins with a detailed description of the proposed ContainerText dataset, followed by an analysis of its data characteristics and challenges. Then we will introduce the annotation method used for this dataset. Finally, we will analyze the widespread use of our proposed dataset in the container industry from a practical application perspective.

A. Dataset Description

The performance of deep learning is significantly impacted by dataset scale and annotation quality. When data are insufficient, a model is prone to overfit. Poor annotation quality will seriously affect the quality of models feature learning. Both of the above situations will severely limit the model performance and its scenario generalization ability. Therefore, high-quality and large-scale datasets are crucial for neural networks. However, at present, a large-scale dataset encompassing annotations for container surface marking text is lacking. This has led to the inability to make progress in deep learning in CMDR and other related fields. Consequently, we have proposed a comprehensive ContainerText dataset at a large scale, along with annotations for text detection and recognition of container markings. The dataset is comprised of 12 K high resolution real world container images with text scripts and bounding box annotations. In order to ensure strong generalization of the dataset, images with various devices, resolutions, and angles were captured and collected. Specifically, this dataset: 1) collected container images under both day and night lighting, 2) used different equipment to capture distorted and undistorted images, 3) captured complete container surfaces and a portion of container surfaces from different angles, and 4) multiple resolutions of images were collected, with a minimum resolution of 1920 $\times$ 1080. Fig. 2 displays examples of the proposed ContainerText dataset. We have annotated texts of different scales on the containers, which mainly comprise numbers, english, punctuation marks, as well as a small number of chinese characters and mathematical notations. All images and annotations have undergone multiple manual cleaning and auditing to ensure their quality. In terms of data characteristics and task challenges, the container's uneven surface causes distortion and occlusion of the text, which poses significant difficulties for text recognition. Examples of the text found in the proposed dataset are displayed in Fig. 3. Additionally, the repetitive textures on the surface of containers cause interference effects on text detection, and like many text detection datasets, ContainerText also has characteristics of multiscaling. Table II analyzes the scales distribution of text instances in the ContainerText dataset. Thus, employing a diverse ContainerText dataset for training neural networks can significantly enhance the model's universality and and its generalization for different usage scenarios.

TABLE II
Scale Distribution of Text Instances in ContainerText Dataset

Scale	Train(Instances)	Test(Instances)	Total(Instances)
Small	17,352	5,473	22,825
Middle	60,134	21,139	81,273
Large	26,876	10,677	37,553

* Instances with a pixel area less than 32^2 are categorized as small-scale, those with an area between 32^2 and 96^2 are categorized as middle-scale, and instances with an area greater than 96^2 are categorized as large-scale.

B. Semi-Automatic Annotation Method

Semi-automatic annotation, assisted by a pretrained model that has undergone training for a similar task, is currently a new method of annotation. Ref. [39] introduced a novel semi-automatic annotation method that leverages active learning and self-paced learning techniques to expedite data labeling and minimize associated expenses. However, for container marking text, the performance of numerous open-source pretraining models currently used in academia is subpar due to the scarcity of comparable training data. For addressing the issue of the current open source models' poor performance on ContainerText, we implemented an incremental learning [40] annotation approach, fully utilizing the new container marking data generated by the labeling process. Fig. 4 outlines the annotation process in detail. Initially, DBNet [41] and SVTR [42] trained on large-scale STR datasets were used to automate annotation. Most of the marking text in the container images will be roughly labeled, but the automatically labeled scripts and bounding boxes still need to be adjusted and filtered. Afterwards, the annotation results undergo manual review and modification. This process requires checking the text of the automatic annotation and the correctness of the bounding box, as well as making corresponding adjustments.

Fig. 2. Example images of ContainerText dataset. We captured over 20 types of containers from different lighting, angles, and resolutions. (a)–(d) were collected in the container stacking area of the factory. (e) and (f) were collected in the production line inside the factory. (g) and (h) were collected in a video of the customs.

Fig. 3. Different container markings in ContainerText. The most prominent feature is the deformation and multi-directional text caused by wavy surfaces. In addition, there are also blurry text caused by small size markings.

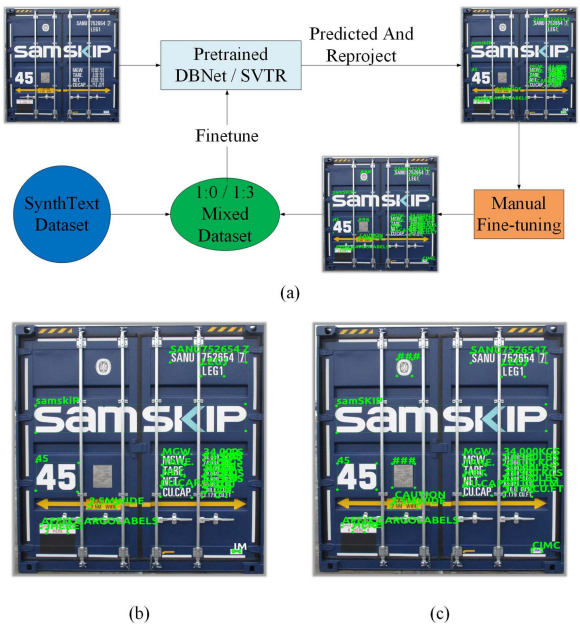


Fig. 4. (a) Flowchart of incremental learning based semi-automatic annotation. (b) The annotation result of pretrained DBNet prediction. (c) The annotation result of manual fine-tuning.

Subsequently, the reviewed data are merged in proportion with the STR datasets, creating a mixed subset for fine-tuning the DBNet. It is important to note that when we combine the newly labeled data with the data use for pretraining in equal proportion

TABLE III
COMPARISON OF ANNOTATION TIME BETWEEN FULLY MANUAL ANNOTATION AND SEMI-AUTOMATIC ANNOTATION

Annotation Method	Average Time	Annotation Accuracy
Fully Manual	5.8 min/image	94.6%
Semi-Automatic(ours)	1.3 min/image	96.3%

for fine tuning, we will freeze the parameters of the model's backbone according to the transfer learning method [43]. The reasons are as follows: 1) by freezing the backbone network, we can preserve the pretraining model's excellent feature extraction ability and prevent catastrophic forgetting [44] of the pretraining knowledge during the fine-tuning process. 2) backbone accounts for most of the parameters of the entire model, and freezing backbone fine-tuning can maximize saving the time-cost and computational resources of the fine-tuning process. This strategy reduces the expense of manual annotation while maximizing efficiency. Table III compares the time cost of fully manual annotation with our proposed semi-automatic annotation method. The mixing ratio experiment will be presented in Section V-C.

C. Downstream Application of ContainerText Dataset

1) Container Object Detection: It was recognized as a fundamental task in the domain of containers for deep learning applications, container object detection aims to accurately locate containers within images while mitigating complex background interference. The proposed ContainerText dataset is an open-world dataset for capturing container images containing a large number of real-world backgrounds. Therefore, the images in the dataset are suitable to both training and evaluating container object detection.

2) Container Damage Detection: Recently, The application of deep learning in container damage detection has also gained significant attention. Ref. [11] proposes an end-to-end method for detecting corrosion defects on container surfaces. However, the dataset they used is not publicly available to other researchers. Our dataset contains container images with varying usage levels, encompassing a wide range of common surface defects including corrosion, damage, and deformation.

TABLE IV
PARAMETERS OF THE CAMERA IN THE CMDR SYSTEM

Parameters	Value
Type	DS-2CD71C7EWD-IZ
Resolution	4,000 × 3,000
Distance to Container	1,800 mm
Field of View	3,650 × 2,300 mm
Exposure	0.04 s

3) Container Lead Seal Detection: For container lead seal problem, many deep learning based methods have been proposed [12], [45]. Container sealing inspections typically focus on detecting the status of lead seals on container doors. The distinctive characteristic of a lead seal is its small size and the similarity in the opening and closing states. The ContainerText dataset contains a substantial collection of images depicting lead seals in various resolutions and states of being open or closed. Thus, our dataset provides a valuable contribution to the advancement of container seal detection tasks.

IV. CONTAINER MARKING DETECTION AND RECOGNITION SYSTEM

In order to achieve fully automated images capture and CMDR tasks, we have built a set of CMDR system, which consist of the CMIAM and the STR based CMDR method. Section IV-A mainly introduces the CMIAM, while STR based CMDR method will be introduced in Section IV-B.

A. Container Marking Image Acquisition Mechanism

The proposed CMIAM is deployed at the final stage of the container production line. Specifically, the CMIAM primarily comprises a steel structure skeleton, 30 cameras (refer to Table IV for detailed camera parameters), a set of adjustable illuminants, and 3 infrared sensors. The steel structure skeleton serves as the main load-bearing carrier of the CMIAM, with thin steel plates laid on the outer side to shield the workshop from the chaotic light. On the inner side, electronic devices including cameras, light sources, and infrared sensors are installed to capture image data from all five surfaces of the container. As the container enters the CMIAM, it halts upon reaching the assigned position, subsequently activating the infrared sensor to emit an electrical signal. Upon receiving the electrical signal, the camera initiates image collection. To account for the container's expansive surface area, we employed 30 cameras to capture some areas of the container surface individually. To ensure marking integrity, each camera possesses a 40 cm of container surface length overlap area in the horizontal direction for photography. Subsequently, once the collection is finalized, the container resumes its forward motion and exits the CMIAM. The gathered images are transmitted to the Data Center, where each image undergoes preprocessing, detection and recognition of marking, and subsequent data storage. The overall operating speed of the proposed system can reach 4.28 FPS (it takes approximately 7 seconds to complete the CMDR process with 30 images), which can achieve real-time detection and recognition of CMDR tasks.

B. STR Based CMDR Method

We have innovatively proposed a STR based method for CMDR tasks. Firstly, we use the position of the text in the marking to replace the position of the marking itself. DBNet++ [29] used as the text detector to detect the position of the text. As for the recognition task, we used SVTR [42] as the text recognizer. Fig. 5 shows the flowchart of STR based CMDR method and Algorithm 1 shows the Pseudo-code.

ResNet [46] uses Feature Pyramid Network (FPN) [47] as the backbone of feature extraction, which is widely used in various deep learning tasks due to its excellent feature extraction capability and multi-scale information fusion ability. To tackle the complex and diverse features present in ContainerText, we employed ResNet-50 in conjunction with a top-down FPN architecture. Specifically, each image $I \in R^{3\times W\times H}$ is first extracted by ResNet-50 to output feature maps of sizes 1/4, 1/8, 1/16 and 1/32 of the original image. In FPN, each layer's feature map undergoes 2 × upsample and is then added by element to the previous layer's feature map for the first multi-scale information fusion.

DBNet++ proposed utilizing the differentiable binarization (1) of probability map to approximate the non differentiable binarization (2), which is integrated into the model training process.

It also replaces conventional fixed binarization thresholds with trainable thresholds map, thereby enhancing the accuracy of post-processing for probability maps. Specifically, the trainable thresholds map is generated by shrinking and dilating the manual labeling bbox with a ratio (which is set to 0.4 empirically) using Vatti Clipping algorithm [48]. After DBNet++ predicts the probability maps, we compute the probability maps and threshold maps according to (1) to obtain approximate binary maps. Approximate binary mapping is used for output text coordinates predicted by the network.

In addition, the Adaptive Scale Fusion Neck fully incorporates contextual information in the feature fusion process. In particular, the feature maps produced by FPN undergo upsampling to achieve a size reduction of 1/4 in relation to the original image. Subsequently, these upsampled feature maps are concatenated according to channel dimensions, obtain an feature map $F \in R^{C\times W\times H}$ and followed by series of convolutions, pooling, activation, dimension expansion, etc., and finally outputs the fused feature map $F \in R^{N\times C\times W\times H}$. Fig. 6(a) presents the details of Adaptive Scale Fusion Neck. The fused feature mapping is used to predict the probability and threshold maps, which are then differentially binarized and subsequently post-processed to generate detection boxes.

$$\hat{B}_{i,j} = \frac{1}{1+e^{-k(P_{i,j}-T_{i,j})}} \tag{1}$$

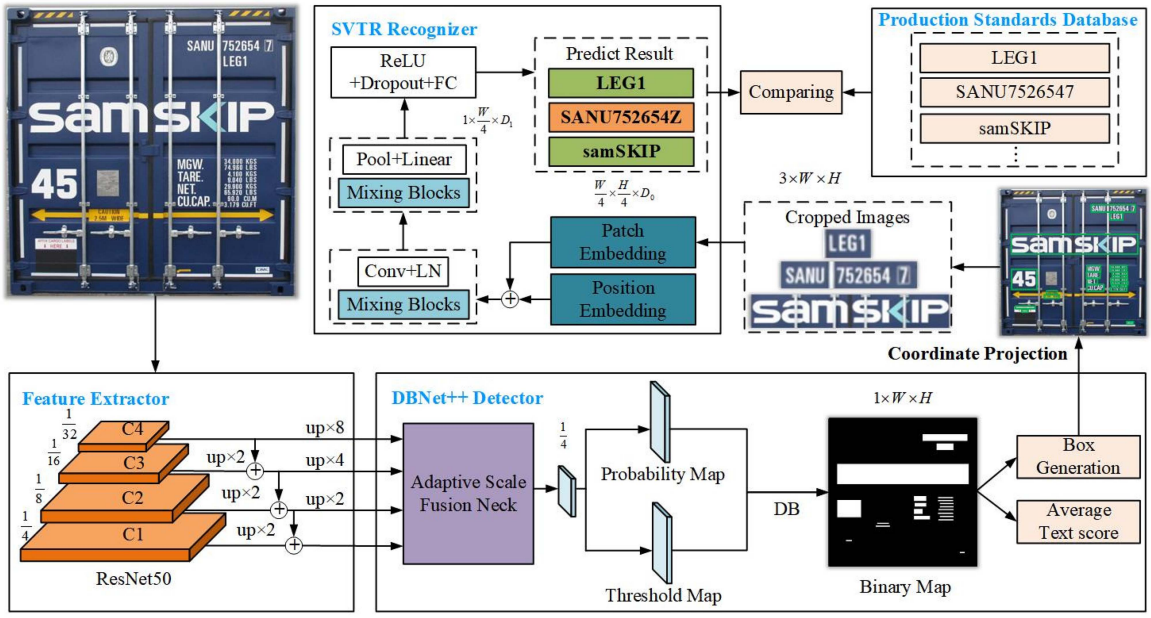


Fig. 5. Flowchart of STR based CMDR method.

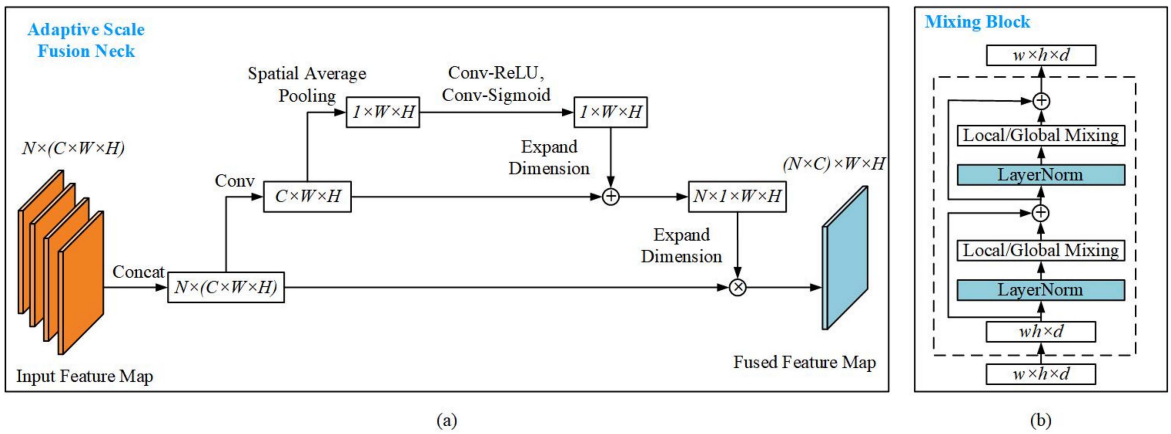


Fig. 6. Details of STR based CMDR method. (a) Adaptive scale fusion neck. (b) Mixing block.

$$B_{i,j} = \begin{cases} 1, & \text{if } P_{i,j} \geq t \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\hat{B}_{i,j}$ is the differentiable binary map, $B_{i,j}$ is the fixed threshold binary map, $P_{i,j}$ is the probability map, t is the fixed threshold, $T_{i,j}$ is learnable threshold map, (i, j) indicates the coordinate point in the map and k is the amplifying factor.

SVTR incorporates the Global / Local Mixing Block to acquire knowledge of both inter-character and inner-character patterns. The details of Mixing Block are presented in Fig. 6(b). The Global Mixing module excels at discriminating between textual and non-textual features, enabling the establishment of long-range dependencies among distinct character features. While the Local Mixing module enhances distinct components within the same character. Furthermore, SVTR operates as a single visual model, simplifying and expediting the feature processing procedure. For the cropped text area, first use convolution and BatchNorm for patch embedding. The encoded patch is added to the positional embedding and then alternately passes through the Mixing Block and convolution followed by LayerNorm. For the output of the last Mixing Block, the height dimension is first pooled to 1, and then the features are compressed into a sequence of features of length D_1 by means of a fully connected layer, an activation layer, and dropout. Finally, the characters are classified using a linear classifier.

Object functions of CMDR are divided into two parts. The first part is the objective function of the detection task, denoted as L_{det}, and the other part is the recognition task, denoted as L_{rec}.

L_{det} is composed of the loss L_s of the probability map, the loss L_t of the threshold map, and the loss L_b of the approximate binary map, respectively.

$$L_{det} = L_s + \alpha \times L_b + \beta \times L_t \tag{3}$$

Algorithm 1: Pseudocode of STR Based CMDR Method.

Input:
Image i of container captured by the CMIAM.
Output:
A List of marking scripts and boxes coordinates.
for all $i \in \{image_1, \ldots, image_{30}\}$ **do**
$detector_{DBNet++}(i)$
$output_{DBNet++}$={[$bbox_1, score_1$],
..., [$bbox_n, score_n$]}
for all $b \in \{bbox_1, \ldots, bbox_n\}$ **do**
Project $bbox_b$ to $image_i$ as $region\ of\ interest$
Crop the $region\ of\ interest$
end for
for all $r \in \{region_1, \ldots, region_n\}$ **do**
$recognizer_{SVTR}(r)$
$output_{SVTR}$={[$scripts_1$], ..., [$scripts_n$]}
end for
return {[$bbox_1, scripts_1$], ..., [$bbox_n, scripts_n$]}
end

α and β are set to 0.1 and 10.

$$L_s = L_b = \sum_{i \in S_l} y_i \log x_i + (1 - y_i) \log(1 - x_i) \tag{4}$$

$$L_t = \sum_{i \in R_d} |y_i^* - x_i^*| \tag{5}$$

S_l is the set of positive and negative samples in the probability map, and we balance the loss based on a 1:3 ratio of positive and negative samples. x_i and y_i represent the predicted result and ground truth of the i-th sample, respectively. R_d is a pixels set of dilated polygon; y^* is the label of the threshold map.

As for L_{rec}, we use CTC loss as the objective function for text recognition.

$$L_{rec} = -\prod_{(x,z) \in S} p(x, z) = \prod_{(x,z)} \ln p(x, z) \tag{6}$$

S is the set of annotation series; x represents the current input sequence, and z represents the output sequence; $p(x, z)$ epresents the probability of the output sequence being z when the input sequence is x.

V. Experiments

We conducted comprehensive experiments on text detection and text recognition algorithms using the ContainerText dataset to establish benchmark performance on the CMDR task. The experimental results can provide performance references for future researchers, comparing their research results with these benchmarks.

A. Experimental Setup

In order to compare fairly, all experiment are implement on Intel Core I5-12600 K CPU, NVIDIA RTX 3060 12 G GPU using python 3.8, pytorch 1.13, CUDA 11.6 with an Ubuntu 18.04. ContainerText has over 12 k images and 141 k instances, and we randomly selected one as the training set for all experiments, which consists of 9 k images and 106 k instances. The remaining images and instances will be used as the testing set. In our text detection experiments, we utilized the Adam optimizer and employed the PolyLR learning rate update strategy, with an initial learning rate of 0.0001. For the text recognition experiment, we utilized an Adam optimizer, linear Warm-Up, and CosineAnnealingLR learning rate update strategy, where we set the incipient learning rate to 0.0003. We have selected suitable data augmentation for each method and trained it to complete convergence. During the testing phase, we did not we did not execute any data augmentation on the images.

B. Evaluation Metric

Similar to most OCR datasets [16], [17], [18], [19], [20], [21], [22], [23], we use Precision (7), Recall (8), and Hmean (9) as evaluation indicators for text detection, while Word Accuracy (10) was used as the evaluation indicator for text recognition. For unrecognizable text, we will mark it as a difficult sample. This difficult samples will participate in the training, but during testing, it will be automatically skipped to avoid affecting the final performance evaluation. Additionally, Frames Per Second (FPS) is used for measuring how fast the model can process an image.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$Hmean = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{9}$$

$$Accuracy = \frac{TP}{N_{total}} \tag{10}$$

Where *TP* is true positive, *TN* is true negative, *FN* is false negative. N_{total} is the total number of test images. In the context of text detection experiments, *TP* is defined as a predicted result with a confidence score exceeding 0.5 and an Intersection over Union ratio with the ground truth surpassing 0.5. For text recognition experiments, *TP* is characterized as the text sequence with the highest prediction confidence, aligning precisely with the ground truth.

C. Semi-Automatic Annotation Mixing Ratio Studies

In the semi-automatic annotation implementation process, fine-tuning the model with newly annotated container data is pivotal for the model's gradual transition from synthetic to container data. To evaluate the impact of the mixing ratio between annotated and synthesized data, we employed the configurations outlined in Section V-A for both the training and testing sets. To simulate newly generated data during annotation, we randomly sampled 300 images / 3300 instances from the training set for a mixed proportion experiment focused on text detection / recognition. Simultaneously, we adjusted the quantity of synthesized data to align with a specific proportion of the sampled container data. The mixing ratio experiments were conducted using DBNet

TABLE V
IMPACT OF DIFFERENT MIXING RATIOS ON THE SEMI-AUTOMATIC ANNOTATION PERFORMANCE OF DBNET

Container (images)	Synthesized (images)	Mixing Ratio	H (%)	Training Time
-	-	only pretrained	39.26	-
300	0	-	**73.15**	0.64 h
300	150	2:1	70.03	0.98 h
300	300	1:1	72.48	1.37 h
300	600	1:2	71.78	2.00 h
300	900	1:3	71.32	2.80 h
300	1500	1:5	70.03	3.99 h

TABLE VI
IMPACT OF DIFFERENT MIXING RATIOS ON THE SEMI-AUTOMATIC ANNOTATION PERFORMANCE OF SVTR

Container (instances)	Synthesized (instances)	Mixing Ratio	Acc. (%)	Training Time
-	-	only pretrained	27.05	-
3300	0	-	60.43	0.48 h
3300	1650	2:1	46.58	0.68 h
3300	3300	1:1	50.28	0.90 h
3300	6600	1:2	65.96	1.35 h
3300	9900	1:3	**67.17**	1.76 h
3300	16500	1:5	66.55	3.36 h

TABLE VII
EXPERIMENTAL RESULTS OF TEXT DETECTION

Method	BS	H(%)	P(%)	R(%)	FLOPs	FPS
TextSnake [49]	2	78.58	85.57	72.64	54.04G	6
PANet [7]	8	66.99	78.68	58.33	49.91G	18
PSENet [51]	8	75.60	80.82	71.02	0.13T	3
DBNet [41]	4	80.28	85.72	75.48	45.96G	**20**
DBNet++ [29]	4	**84.41**	**87.45**	**81.57**	61.09G	10

and SVTR, and the results are presented in Tables V and VI (the *Mixing Ratio* in the table denotes the ratio of newly annotated data to synthesized data). By comparing *Hmean* or *Accuracy* and *Training Time* across different mixing ratios, it was conclusively demonstrated that 1:0 and a 1:3 mixing ratio are the most efficient proportions for DBNet and SVTR, respectively.

D. Comparative Studies

Given the excellent performance of deep learning in various vision tasks, a great deal of deep learning based approaches have been proposed to address the challenges in the STR domain. We have selected some methods in recent years and conducted numerous experiments as the benchmark performance for the ContainerText dataset. The results of our experiments are shown in Tables VII and VIII. To analyze the performance of the selected approaches on the ContainerText dataset, we visualize the experimental results for the text detection and text recognition tasks separately.

TABLE VIII
EXPERIMENTAL RESULTS OF TEXT RECOGNITION

Method	Backbone	BS	Acc.(%)	FLOPs	FPS
Aster [52]	Res-22	64	86.25	0.92G	**941**
NRTR [33]	Res-31	64	88.39	-	171
SAR [34]	Res-31	64	87.57	33.54G	36
SATRN [53]	ConvLayer	32	**88.49**	-	52
RobustScanner [54]	Res-31	64	87.35	-	213
ABINet [55]	Res-ABI	64	87.45	8.34G	297
SVTR [42]	MixingBlock	64	86.76	4.16G	377

1) Text Detection: During the experiment on text detection, we adjusted the batch size of each model to a suitable maximum value for accelerating the training process. We utilized a ResNet50, which pretrained on ImageNet, as the initialization weight for each model's backbone network. Among the different models tested, DBNet++ exhibited superior performance in the aspect of Hmean, Precision, and Recall, achieving respective values of 84.17%, 87.64%, and 80.96%. In the aspect of speed, it reaches a rate of 10 FPS, second only to DBNet's 20 FPS. A distinguishing feature of DBNet++ is the integration of the Adaptive Scale Fusion Neck instead of directly concatenating feature maps, resulting in its significant superiority over DBNet in *Hmean* and *Recall* metrics. Compared with alternative methods, DBNet++ excels in accurately delineating text region boundaries, attributed to its application of differentiable binarization. Visualization of prediction results from various methods in the comparative studies is presented in Fig. 7.

2) Text Recognition: In the text recognition experiments, SATRN achieved the highest accuracy of 88.49%, which surpasses Aster, the lowest accuracy, of 86.25% by 2.24%. Notably, Aster proved to be the fastest model, operating at 941 FPS, which is nearly 20 times faster than the slowest SAR model. To meet the real-time demands of CMDR tasks, SVTR strikes a better balance between speed and accuracy, achieving an accuracy of 86.76% and operating at 377 FPS. In addition, we intentionally rewrite the annotation information of some mathematical notation in the dataset into another format (refering to the last column in Fig. 8 for details), and then use this part of the data for training and testing. Surprisingly, nearly all models exhibit the ability to adjust to this rewriting format. The experiment demonstrates that text recognition models prioritize the textual patterns present within images and exhibit a good adaptability to such patterns. Consequently, it becomes feasible to substitute intricate text with simple symbols, thereby simplifying the processing of complex textual data during annotation or post-processing.

E. Ablation Studies

Due to their excellent performance in the aspect of speed and performance, we chose DBNet++ and SVTR as the main detection and recognition models for the CMDR task. We conducted a series of ablation experiments specifically targeting DBNet++ and SVTR. The objective of these experiments was to investigate the impact of adjusting model parameters within a consistent

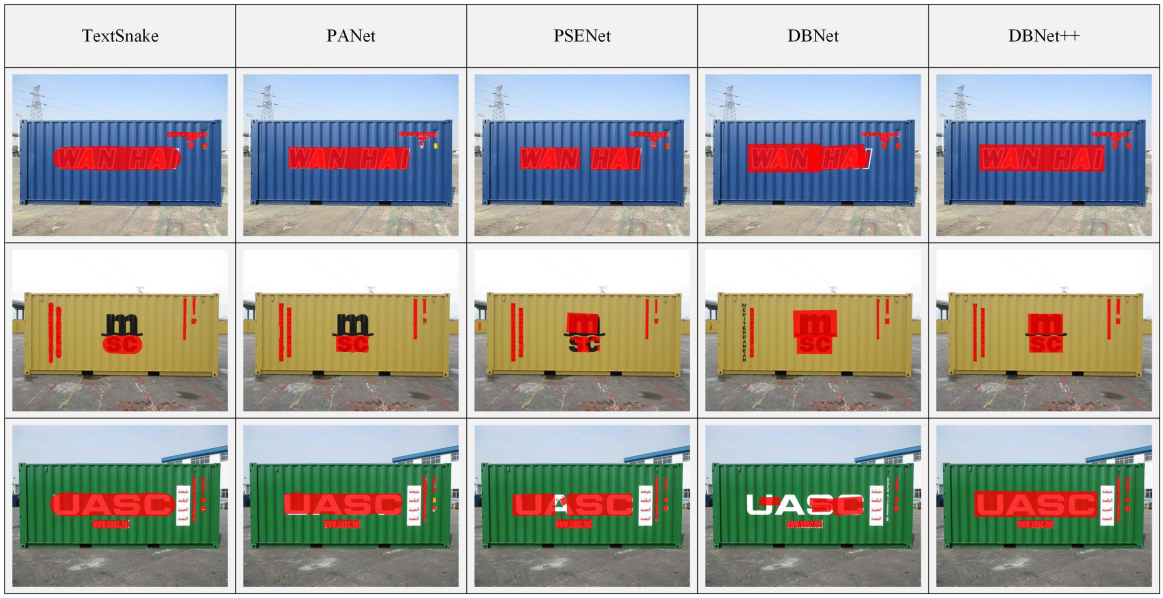


Fig. 7. Text detection results. The red mask is the text area predicted by the model.

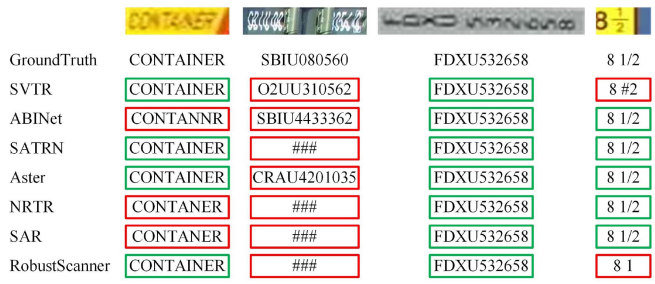


Fig. 8. Text recognition results. "#" represents that the model cannot recognize the character in the image. The green box represents correct recognition, while the red box represents incorrect recognition.

structure on performance, with the goal of achieving an optimal trade-off between inference speed and performance.

1) Text Detection: The backbone of DBNet++ contributes the main parameters. We conducted training on the backbone using different parameters of DBNet++. In addition, to assess the performance difference between the Transformer and CNN backbones, we incorporated the Swin Transformer V2 [57] series as the backbone for our experimental models. When employing ResNet-18 as the backbone network for DBNet++, false positives in the prediction results were observed, indicating overlaps with other prediction boxes. This suggests that the discriminative power of features extracted by ResNet-18 is inferior to deeper and larger networks such as ResNet-50 and Swin Transformer. Conversely, when utilizing ResNet-50+DCN and Swin-T as backbone networks, the model's prediction results did not entirely cover the text area, resulting in some text edges extending beyond the detection box. Fig. 9 displays the prediction results of the model using different backbones. The experiment illustrates that employing ResNet-50 as the backbone yields the highest overall performance, with an *Hmean* of 84.17% and *FPS* of 10. The experimental results are presented in Table IX.

TABLE IX
ABLATION EXPERIMENT RESULTS OF DBNET++

Backbone	Params	H(%)	P(%)	R(%)	FPS
Res-18	12.97M	81.89	89.98	75.13	**13**
Res-50	26.03M	84.41	87.45	81.57	10
Res-50 + DCN [56]	26.91M	83.71	88.96	79.04	8.5
Swin-T [57]	29.43M	82.74	87.70	78.32	6.4

TABLE X
HYPER-PARAMETER STUDY OF α AND β

α	β	Hmean(%)	Precision(%)	Recall(%)
0.1	0.1	83.94	86.52	81.52
0.1	0.5	83.78	85.37	82.25
0.1	1	84.31	86.14	82.55
0.1	5	83.76	85.48	82.11
0.1	10	84.41	87.45	81.57
0.5	10	83.92	85.32	82.57
1	10	83.89	86.38	81.54
5	10	83.92	85.53	82.36
10	10	83.70	85.42	82.06

In addition to assessing the backbone network, we conducted extensive ablation experiments to explore the impact of varying values for α, β in the objective function and k in (1) of DBNet++. Due to the numerous combinations of α and β values, we initiated the experiments by randomly selecting 0.1 as the initial value for α and testing optimal values for β. Subsequently, with a fixed β, we tested diverse values for α. Upon determining the optimal combination of α and β, we employed that setting for k experiments. Through experimentation, optimal values for α, β, and k were identified as 0.1, 10, and 50, respectively. The detailed experimental results are available in Tables X and XI.

Fig. 9. Prediction results of the model using different backbones. The red mask is the text area predicted by the model.

TABLE XI
HYPER-PARAMETER STUDY OF K

k	Hmean(%)	Precision(%)	Recall(%)
0.1	83.91	85.74	82.15
0.5	83.99	85.84	82.22
1	83.97	85.27	82.72
10	84.02	87.68	80.66
50	84.41	87.45	81.57

TABLE XII
ABLATION EXPERIMENT RESULTS OF SVTR

Local	Global	Params	Accuracy(%)	FPS
6	6	4.15M	81.34	**1130**
8	7	8.45M	83.52	1027
8	10	22.66M	**86.76**	377
10	11	38.81M	83.83	251

TABLE XIII
DATASET SCALE EXPERIMENTS RESULTS OF DBNET++

Method	Backbone	Params	H(%)		
			10%	30%	60%
DBNet++	Res-18	12.97M	71.59	78.85	80.68
DBNet++	Res-50	26.03M	74.39	81.50	82.31

2) Text Recognition: The primary parameters of the SVTR model are from Local / Global Mixing Blocks. Therefore, we control the total parameters of SVTR by changing the number of Local / Global Mixing Blocks. The experimental results present that when the quantity of Local / Global Mixing Blocks is both 6, the model has the least quantity of parameters, only 4.15 million. At this point, the model reached 1130 FPS, but the accuracy was only 81.34%. When the Local / Global Mixing Blocks are 8 and 10, respectively, the model parameters is 22.66 million. The accuracy is 86.76%, and the FPS is 377, reaching a balance between speed and performance. The experimental results are presented in Table XII.

F. Dataset Scale Effecct Studies

The matching relationship between neural network parameters and the number of training data has always been a hot topic for deep learning researchers. We use DBNet++ and SVTR as representative models for text detection and recognition, and explore the training data required for CMDR tasks on the ContainerText dataset. We continue to use the sampling results from Section V-A regarding the training and validation sets. We randomly select 10%, 30%, and 60% data from the original training set each time for small sample experiments to verify the matching relationship between the training data and the neural network. We set 80% Hmean or Accuracy as the performance baseline. The results of small sample experiments for detection and recognition tasks are listed in Tables XIII and XIV.

In detection tasks, the most important evaluation indicator Hmean shows a positive correlation trend with model parameters and training data. When the number of model parameters is smaller, more training data is needed to achieve the expected performance. In addition, we also observed that whether using

TABLE XIV
DATASET SCALE EXPERIMENTS RESULTS OF SVTR

Method	Backbone	Params	Accuracy(%)		
			10%	30%	60%
SVTR	Local 6 + Global 6	4.15M	45.37	59.79	79.82
SVTR	Local 8 + Global 10	22.66M	29.09	49.46	81.19
SVTR	Local 10 + Global 11	38.81M	32.51	54.94	81.67

models with large or small parameters, the growth rate of model performance gradually decreases with the increase of training data.

However, in recognition tasks, the situation between parameters and training data becomes complex. Overall, the performance of the model increases with the increase of training data. But when the training data is less than 30% (i.e. 42000 instances), changes in model parameters can lead to unpredictable changes in model performance.

G. Limitation

In this study, we introduced the most extensive container marking text dataset and corresponding semi-automatic annotation methods. Simultaneously, we implemented a fully automated CMIAM. This module, when used in conjunction with the proposed STR based CMDR method, has facilitated the achievement of high-performance container text recognition. Despite demonstrating outstanding performance in container text detection and recognition tasks, it is noteworthy that the text detection and recognition models are optimized using non-end-to-end methods. This optimization approach entails the risk of error accumulation and sub-optimal issues during the optimization process. Furthermore, the use of non-end-to-end detection and recognition models necessitates distinct data inputs and preprocessing steps, thereby introducing complexity to the data processing within the detection system.

VI. CONCLUSION

In this paper, we put forward a complete CMDR system that including an open large-scale container marking text dataset (called ContainerText) for CMDR tasks, a STR-based CMDR method and an Container Marking Image Acquisition Mechanism (CMIAM). The proposed ContainerText dataset has more than 1,2000 high-resolution images, providing coordinates and text content for container markings. To address the problem of costly manual annotation, we propose a semi-automatic annotation method combining incremental learning and transfer learning, which saves annotation costs and improves annotation accuracy. Secondly, the STR based CMDR method is proposed for fine-grained container marking recognition. We carried out substantial feasibility analysis experiments on the proposed ContainerText dataset and constructed a STR based CMDR benchmark. The experimental results present that our suggested method can fully adapt to CMDR tasks. In addition, we also use electrical components such as cameras, infrared sensors, and light sources to form CMIAM for automated high-quality container marking image acquisition and CMDR task. All experimental data and CMIAM parameters of this study will be made public, providing reference for future researchers.

VII. FUTURE WORK

While the current SOTA model for text detection and recognition achieves commendable accuracy in CMDR tasks, there remains a need to enhance the model's efficacy in addressing surface irregularities and repetitive textures on containers. It is imperative to propose an enhanced CMDR model capable of navigating such challenges for more accurate CMDR task completion. Furthermore, adopting a straightforward training and deployment approach is crucial when integrating deep learning into CMDR tasks. However, when employing STR for CMDR tasks, there is still a requirement to deploy text detection and text recognition models separately. Consequently, investigating the development of an end-to-end CMDR model stands as a noteworthy research direction for the future.

REFERENCES

[1] Y. Bengio, Y. Lecun, and G. Hinton, "Deep learning for AI," *Commun. ACM*, vol. 64, no. 7, pp. 58–65, 2021.
[2] A. Pramanik, S. K. Pal, J. Maiti, and P. Mitra, "Granulated RCNN and multi-class deep SORT for multi-object detection and tracking," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 1, pp. 171–181, Feb. 2022.
[3] H. Wang, Y. Xu, Z. Wang, Y. Cai, L. Chen, and Y. Li, "CenterNet-Auto: A multi-object visual detection algorithm for autonomous driving scenes based on improved CenterNet," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 742–752, Jun. 2023.
[4] S. P. Sahoo, S. Ari, K. Mahapatra, and S. P. Mohanty, "HAR-Depth: A novel framework for human action recognition using sequential learning and depth estimated history images," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 5, pp. 813–825, Oct. 2021.
[5] X. Qu et al., "Attend to where and when: Cascaded attention network for facial expression recognition," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 3, pp. 580–592, Jun. 2022.
[6] Y. Zhang et al., "A quantum-inspired multimodal sentiment analysis framework," *Theor. Comput. Sci.*, vol. 752, pp. 21–40, 2018.
[7] S. Yang, D. Zhou, J. Cao, and Y. Guo, "LightingNet: An integrated learning method for low-light image enhancement," *IEEE Trans. Comput. Imag.*, vol. 9, pp. 29–42, 2023.
[8] S. Yang, D. Zhou, J. Cao, and Y. Guo, "Rethinking low-light enhancement via Transformer-GAN," *IEEE Signal Process. Lett.*, vol. 29, pp. 1082–1086, 2022.
[9] Y. Guo, D. Zhou, P. Li, C. Li, and J. Cao, "Context-aware Poly(A) signal prediction model via deep spatial—temporal neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, Dec. 2022.
[10] Y. Guo, D. Zhou, X. Ruan, and J. Cao, "Variational gated autoencoder-based feature extraction model for inferring disease-miRNA associations based on multiview features," *Neural Netw.*, vol. 165, pp. 491–505, 2023.
[11] Z. Bahrami, R. Zhang, T. Wang, and Z. Liu, "An end-to-end framework for shipping container corrosion defect inspection," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 5020814.
[12] R. Cao, Q. Liu, and G. Zhang, "An algorithm for container lead sealing detection," *SN Comput. Sci.*, vol. 4, no. 4, 2023, Art. no. 412.
[13] Z. Bahrami, R. Zhang, R. Rayhana, and Z. Liu, "An HRCR-CNN framework for automated security seal detection on the shipping container," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5018113.
[14] R. Zhang, Z. Bahrami, and Z. Liu, "A vertical text spotting model for trailer and container codes," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5017313.
[15] R. Zhang, Z. Bahrami, T. Wang, and Z. Liu, "An adaptive deep learning framework for shipping container code localization and recognition," *IEEE Trans. Instrum. Meas.*, vol. 70, 2020, Art. no. 2501013.
[16] D. Karatzas et al., "ICDAR 2013 robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit.*, 2013, pp. 1484–1493.
[17] D. Karatzas et al., "ICDAR 2015 competition on robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit.*, 2015, pp. 1156–1160.

[18] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2315–2324.
[19] C. Ch'ng, C. Chan, and C.-L. Liu, "Total-text: Toward orientation robustness in scene text detection," *Int. J. Document Anal. Recognit.*, vol. 23, pp. 31–52, 2020.
[20] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognit.*, vol. 90, pp. 337–345, 2019.
[21] A. Singh, G. Pang, M. Toh, J. Huang, W. Galuba, and T. Hassner, "TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8802–8812.
[22] N. Nayef et al., "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification - RRC-MLT," in *Proc. Int. Conf. Document Anal. Recognit.*, 2017, pp. 1454–1459.
[23] N. Nayef et al., "ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition - RRC-MLT-2019," in *Proc. Int. Conf. Document Anal. Recognit.*, 2019, pp. 1582–1587.
[24] Y. Su et al., "TextDCT: Arbitrary-shaped text detection via discrete cosine transform mask," *IEEE Trans. Multimedia*, vol. 25, pp. 5030–5042, 2023.
[25] Z. Shao et al., "CT-Net: Arbitrary-shaped text detection via contour transformer," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 3, pp. 1815–1826, Mar. 2024, doi: 10.1109/TCSVT.2023.3299087.
[26] Y. Su et al., "LRANet: Towards accurate and efficient scene text detection with low-rank approximation network," 2023, *arXiv:2306.15142*.
[27] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4159–4167.
[28] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6773–6780.
[29] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 919–931, Jan. 2023.
[30] C. Zhai, Z. Chen, J. Li, and B. Xu, "Chinese image text recognition with BLSTM-CTC: A segmentation-free method," in *Proc. Chin. Conf. Pattern Recognit.*, 2016, pp. 525–536.
[31] M. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.
[32] A. Graves, F. Santiago, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 369–376.
[33] F. Sheng, Z. Chen, and B. Xu, "NRTR: A no-recurrence sequence-to-sequence model for scene text recognition," in *Proc. Int. Conf. Document Anal. Recognit.*, 2019, pp. 781–786.
[34] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8610–8617.
[35] T. Zheng, Z. Chen, S. Fang, H. Xie, and Y. Jiang, "Cdistnet: Perceiving multi-domain character distance for robust text recognition," *Int. J. Comput. Vis.*, vol. 132, no. 2, pp. 300–318, 2024.
[36] W. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin, "GTC: Guided training of CTC towards efficient and accurate scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11005–11012.
[37] Y. Wang, H. Xie, S. Fang, J. Wang, S. Zhu, and Y. Zhang, "From two to one: A new scene text recognizer with visual language modeling network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 14174–14183.
[38] Y. Du et al., "Context perception parallel decoder for scene text recognition," 2023, *arXiv:2307.12270*.
[39] P. Zhu, Y. Sun, B. Cao, X. Liu, X. Liu, and Q. Hu, "Self-supervised fully automatic learning machine for intelligent retail container," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 5021215.
[40] S. A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5533–5542.
[41] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11474–11481.
[42] Y. Du et al., "SVTR: Scene text recognition with a single visual model," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 5533–5542.
[43] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*.
[44] R. French, "Catastrophic forgetting in connectionist networks," *Trends Cogn. Sci.*, vol. 3, no. 4, pp. 128–135, 1999.
[45] G. Zhang, J. Guo, Q. Liu, and H. Wang, "Container lead seal detection based on Nano-CenterNet," in *Proc. Int. Conf. Neural Comput. Adv. Appl.*, 2022, pp. 210–221.
[46] 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.
[47] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.
[48] B. R. Vati, "A generic solution to polygon clipping," *Commun. ACM*, vol. 35, no. 7, pp. 56–64, 1992.
[49] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A flexible representation for detecting text of arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 19–35.
[50] W. Wang et al., "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8439–8448.
[51] W. Wang et al., "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9328–9337.
[52] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, Sep. 2019.
[53] J. Lee, S. Park, J. Baek, S. Oh, S. Kim, and H. Lee, "On recognizing texts of arbitrary shapes with 2D self-attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 2326–2335.
[54] X. Yue, Z. Kuang, C. Lin, H. Sun, and W. Zhang, "RobustScanner: Dynamically enhancing positional clues for robust text recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 135–151.
[55] 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 Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7094–7103.
[56] R. Wang et al., "DCN V2: Improved deep & cross network and practical lessons for web-scale learning to rank systems," in *Proc. Web Conf.*, 2021, pp. 1785–1797.
[57] Z. Liu et al., "Swin transformer V2: Scaling up capacity and resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11999–12009.

Ying Xu received the Ph.D. degree in automation from the South China University of Technology, Guangdong, China, in 2013. She is currently an Associate Professor with the School of Electronics and Information Engineering, Wuyi University, Jiangmen, China. Her current research interests include image processing, deep learning, biometric extraction, and optical character recognition.

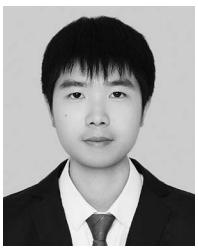
Zhangzhao Liang received the B.S. degree from the Zhongkai University of Agriculture and Engineering, Guangzhou, China, in 2019. He is currently working toward the master's degree with the School of Electronics and Information Engineering, Wuyi University, Nanping, China. His research interests include computer vision, optical character recognition, and image processing.

Yanyang Liang received the Ph.D. degree in automation from the University of Science and Technology of China, Hefei, China, in 2008. He is currently an Associate Professor with the School of Electronics and Information Engineering, Wuyi University, Jiangmen, China. His current research interests include: computer vision, motion controls, and robust control for nonlinear systems.

Xinru Li received the B.S. degree from Guangdong Ocean University, Zhanjiang, China, in 2022. She is currently working toward the master's degree with the School of Electronics and Information Engineering, Wuyi University, Nanping, China. Her research interests include computer vision, scene text detection, and image processing.

Wenfeng Pan received the B.S. degree from Wuyi University, Nanping, China, in 2019. He is currently working toward the master's degree with the School of Electronics and Information Engineering, Wuyi University. His research interests include computer vision, pattern recognition, and image processing.

Jie You received the B.S. degree from Wuyi University, Nanping, China, in 2019. He is currently working toward the master's degree with the School of Electronics and Information Engineering, Wuyi University. His research interests include: computer vision, optical character recognition, and image processing.



Zhihao Long received the B.S. degree from Wuyi University, Nanping, China, in 2022. He is currently working toward the master's degree with the School of Electronics and Information Engineering, Wuyi University. His research interests include computer vision, image generation, and image processing.

Yikui Zhai (Senior Member, IEEE) received the Ph.D. degree in signal and information processing from Beihang University, Beijing, China, in June 2013. Since October 2007, he has been with Wuyi University, Jiangmen, China, where he is currently a Full Professor. Since 2021, he has also been the Associate Dean of the School of Electronics and Information Engineering, Wuyi University, Nanping China. He was a Visiting Scholar with the Department of Computer Science, Università degli Studi di Milano, Milan, Italy, from 2016 to 2017, August 2023 and January 2024. His research interests include image processing, deep learning, optical character recognition, object detection, UAV change detection, self-supervise learning.

Angelo Genovese (Senior Member, IEEE) received the Ph.D. degree in computer science from the Università degli Studi di Milano, Milan, Italy, in 2014. Since 2022, he has been an Associate Professor in computer science with the Università degli Studi di Milano. He was a Visiting Researcher with the University of Toronto, Toronto, ON, Canada. Original results have been authored or coauthored in more than 30 papers in international journals, proceedings of international conferences, books, and book chapters. His current research interests include Signal and Image processing, three-dimensional reconstruction, computational intelligence technologies for biometric systems, industrial and environmental monitoring systems, and design methodologies and algorithms for self-adapting systems. Dr. Genovese is an Associate Editor for the *Journal of Ambient Intelligence and Humanized Computing* (Springer).

Vincenzo Piuri (Fellow, IEEE) received the Ph.D. degree in computer engineering from the Politecnico di Milano, Milan, Italy, in 1989. He has been a Full Professor of computer engineering with the Università degli Studi di Milano, since 2000. He has also been an Associate Professor with the Politecnico di Milano, Italy, and a Visiting Professor with The University of Texas at Austin, Austin, TX, USA, and a Visiting Researcher with George Mason University, Fairfax, VA, USA. He is currently an Honorary Professor with Obuda University, Budapest, Hungary, Guangdong University of Petrochemical Technology, Maoming, China, Northeastern University, Shenyang, China, Muroran Institute of Technology, Muroran, Japan, and Amity University, Noida, India. Original results have been authored or coauthored in more than 400 articles in international journals, proceedings of international conferences, books, and book chapters. His research interests include: artificial intelligence, computational intelligence, intelligent systems, machine learning, pattern analysis and recognition, signal and image processing, biometrics, intelligent measurement systems, industrial applications, digital processing architectures, fault tolerance, dependability, and cloud computing infrastructures. Dr. Piuri is also a Distinguished Scientist of ACM, an ACM Fellow and a Senior Member of INNS. He has been President of the IEEE Systems Council during 2020–2021, and IEEE Vice President for Technical Activities in 2015, IEEE Director, President of the IEEE Computational Intelligence Society, Vice President for Education of the IEEE Biometrics Council, Vice President for Publications of the IEEE Instrumentation and Measurement Society and the IEEE Systems Council, and Vice President for Membership of the IEEE Computational Intelligence Society.

Fabio Scotti (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the Politecnico di Milan, Milan, Italy, in 2003. He is currently a Full Professor with the Università degli Studi di Milano, Milan, Italy, since 2020. His original results have been authored or coautored in more than 130 papers in international journals, proceedings of international conferences, books, book chapters, and patents. His current research interests include: biometric systems, machine learning and computational intelligence, signal and image processing, theory and applications of neural networks, three-dimensional reconstruction, industrial applications, intelligent measurement systems, and high-level system design. Dr. Scotti is an Associate Editor with IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS and IEEE OPEN JOURNAL OF SIGNAL PROCESSING. He is the Book Editor (Area Editor, section Less-constrained Biometrics) of the*Encyclopedia of Cryptography, Security, and Privacy* (3 rd Edition), Springer. He has been an Associate Editor with IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, *Soft Computing* (Springer) and a Guest Co-editor for IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.