Optimal Boxes: Boosting End-to-End Scene Text Recognition by Adjusting Annotated Bounding Boxes via Reinforcement Learning

Jingqun Tang[1]*, Wenming Qian[2]*, Luchuan Song[3], Xiena Dong[4], Lan Li[5], and Xiang Bai[6]✉

[1] Ant Group
jingquntang@163.com
[2] NetEase Fuxi AI Lab
wenmingqian@corp.netease.com
[3] University of Rochester
lsong11@ur.rochester.edu
[4] Hangzhou Dianzi University
dxn@hdu.edu.cn
[5] Wuhan University
2016302580090@whu.edu.cn
[6] Huazhong University of Science and Technology
xbai@hust.edu.cn

Abstract. Text detection and recognition are essential components of a modern OCR system. Most OCR approaches attempt to obtain accurate bounding boxes of text at the detection stage, which is used as the input of the text recognition stage. We observe that when using tight text bounding boxes as input, a text recognizer frequently fails to achieve optimal performance due to the inconsistency between bounding boxes and deep representations of text recognition. In this paper, we propose Box Adjuster, a reinforcement learning-based method for adjusting the shape of each text bounding box to make it more compatible with text recognition models. Additionally, when dealing with cross-domain problems such as synthetic-to-real, the proposed method significantly reduces mismatches in domain distribution between the source and target domains. Experiments demonstrate that the performance of end-to-end text recognition systems can be improved when using the adjusted bounding boxes as the ground truths for training. Specifically, on several benchmark datasets for scene text understanding, the proposed method outperforms state-of-the-art text spotters by an average of 2.0% F-Score on end-to-end text recognition tasks and 4.6% F-Score on domain adaptation tasks.

Keywords: End-to-End Text Recognition, Reinforcement Learning, Optimal Bounding Boxes

* Equal contribution. ✉ Corresponding author.

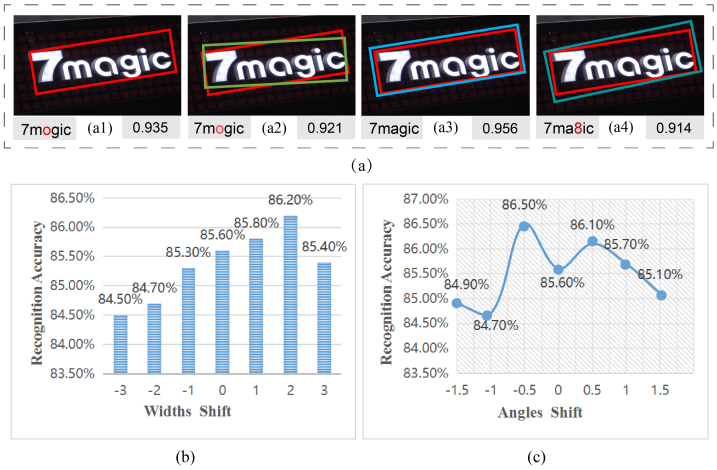

Fig. 1. (a) The red boxes represent the ground-truth bounding boxes, while the others are randomly shifted. Fig.(a1) represents the recognition confidence and recognition results with the ground-truth bounding box, while Fig.(a2) to Fig.(a4) with randomly shifted bounding boxes. The recognition results are presented on the left of (a1) to (a4), and the recognition confidence on the right; (b) text recognition accuracy with adjusting widths of the ground-truth bounding boxes; (c) text recognition accuracy with adjusting angles of the ground-truth bounding boxes.

1 Introduction

In modern society, text plays a more important role than ever before as an essential tool for communication and collaboration. Meanwhile, scene text reading has become an active research area due to its wide applications in the real world, such as image instant translation [8,39], image search [36,41], and industrial automation [1,12].

Text detection and recognition can be roughly divided into two categories: two-step systems and end-to-end systems. For two-step systems [45,24,2,40,37,38,21], since detected texts are cropped from the image, detection and recognition are two separate steps. Some of these methods first generate text proposals using a text detection model and then recognize them with a text recognition model [13,23,9]. For end-to-end systems, many end-to-end trainable networks [3,4,11,20,27] have recently been proposed. [4,11,27] develop unified text detection and recognition systems with very similar overall architectures, which consist of a recognition branch and a detection branch. However, current models simply use tight annotated text bounding boxes as the ground truth, ignoring the inconsistency between bounding boxes and deep representations of text recognition. So, are tight bounding boxes the most suitable for recognition tasks? Through a series of experiments, we observe that a text recognizer frequently fails to achieve its best performance when using tight bounding boxes as inputs.

As shown in Fig.1(a), with suitable adjustments to the bounding boxes, we can get higher recognition confidence and correct recognition results (see Fig.1(a3)). As shown in Fig.1(b) and Fig.1(c), the text recognizer can perform better when adjusting the widths or rotation angles of the ground-truth bounding boxes. The above experiments show a certain inconsistency between bounding boxes and deep representations of text recognition. Additionally, unlike in COCO [25], where clipping two pixels off an object does not prevent recognition, a 1-2 pixel error in text boxes may render the correct recognition prediction unrecoverable. The text recognition result is more sensitive to changes in the bounding box. To address the aforementioned problems, this paper presents a reinforcement learning-based method for adjusting the shape of each ground-truth bounding box so that it is more compatible with the text recognition task.

We propose a reinforcement learning-based method named Box Adjuster, which mitigates the inconsistency between bounding boxes and deep representations of text recognition. Our method can be summarized as follows: Firstly, we choose a range of representative text recognizers and regard the average recognition confidence as a reward. Secondly, the **Box** Adjusting **D**eep **Q** **N**etwork (BoxDQN) with Feature Fusion Module (FFM) is trained, which can automatically adjust bounding boxes according to the text recognition reward. Finally, we train the end-to-end scene text recognition model with the refined ground-truth bounding boxes for better recognition. Furthermore, as a preprocessing method, it is only applied in the process of creating training datasets. Thus, there is no additional computational cost in the forward phase.

Additionally, the proposed Box Adjuster is beneficial for resolving cross-domain problems such as synthetic-to-real, in which the source domain represents labeled synthetic data and the target domain represents unlabeled real data. To prove the effectiveness and generalization of our approach, we conduct experiments on standard benchmarks, including ICDAR 2013 [17], ICDAR 2015 [16], ICDAR 19-ReCTS [44] and ICDAR 19-MLT [32] datasets. The proposed method achieves better performance on the datasets when compared with the existing state-of-the-art methods. Besides, we demonstrate the efficacy of our approach on domain adaptation tasks.

Our contributions can be summarized as follows:

- We introduce the Box Adjuster, which adjusts the shape of each annotated text bounding box to make it more compatible with text recognition models. Besides, a text recognition-based reward is proposed to train our BoxDQN model in order to capture optimal annotated bounding boxes.
- Our proposed Feature Fusion Module (FFM), which integrates foreground, background, and box coordinates, considerably enhances BoxDQN in terms of application scope and accuracy.
- Our approach is generalized and can be easily applied to boost existing OCR systems without any additional computational cost during the inference phase. Concurrently, the proposed method outperforms state-of-the-art text spotters by an average of 2.0% F-Score on public datasets.

– When utilized in the cross-domain area, the proposed method significantly mitigates inconsistency between source and target domains, resulting in an average improvement of 4.6% for state-of-the-art text spotters.

2 Related Works

2.1 Two-Step OCR Systems

In two-step systems, due to the fact that detected texts are cropped from the image, the detection and recognition are two separate steps. Some of these methods first generate text proposals using a text detection model [45,24,40] and then recognize them with a text recognition model [13,23,9]. Jaderberg et al. [13] use a combination of Edge Box proposals [46] and a trained aggregate channel features detector [7] to generate candidate text bounding boxes. Liao et al. [23] combine an SSD [26] based text detector and CRNN [37] to spot text in images. In addition, for the detection step, EAST [45] further simplifies the anchor-based detection by adopting the U-shaped design [35] to integrate features from different levels. And for the recognition step, RARE [38] consists of a Spatial Transformer Network (STN) and a Sequence Recognition Network (SRN), which is robust to irregular text. One major disadvantage of two-step methods is that the propagation of error between the recognition and detection models will result in less satisfactory performance.

2.2 End-to-End OCR Systems

Many end-to-end trainable networks have recently been proposed [3,4,20,11,27]. Bartz et al. [3] present a solution that employs a STN [14] to attend to each word in the input image circularly and then recognize them individually. Li et al. [20] substitute the object classification module in Faster-RCNN [34] with an encoder-decoder-based text recognition model and to create their text spotting system [4,11,27] develop unified text detection and recognition systems with very similar overall architectures, which consist of a recognition branch and a detection branch. Liu et al. [28] design a novel BezierAlign layer for extracting accurate convolution features of a text instance with arbitrary shapes and adaptively fit arbitrarily-shaped text via a parameterized Bezier curve. Liao et al. [22] propose Mask Text Spotter v3, an end-to-end trainable scene text spotter that adopts a Segmentation Proposal Network (SPN) instead of an RPN [34].

2.3 Reinforcement Learning

In earlier work, Mnih et al. [30] present the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. In recent years, reinforcement learning [18,15,5,29] has evolved considerably in the field of object detection. Some works [33,43] employ reinforcement learning as a post-processing method for scene text detection

to adjust the bounding boxes predicted by the detection model, which can result in a significant time increase. In contrast to earlier research, we employ text recognition as the reward rather than the IOU between the predicted and ground-truth bounding boxes. In addition, our method is not a post-processing of the detection and does not add any extra computational costs to the inference phase.

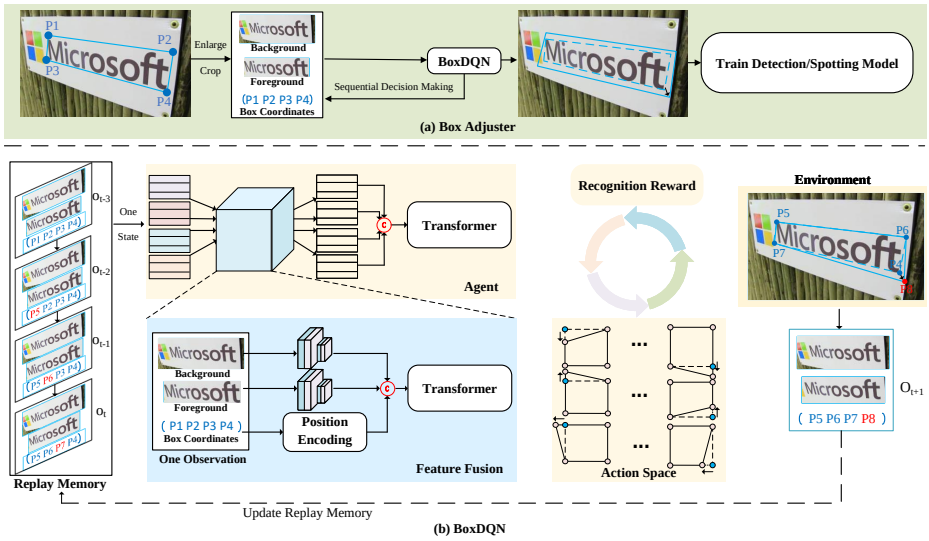

Fig. 2. Overview of our proposed method Box Adjuster and the details of BoxDQN model architecture. In order to mitigate the inconsistency between bounding boxes and deep representations of text recognition, we utilize Box Adjuster to adjust ground-truth bounding boxes and train the text spotter with them. BoxDQN is a method based on reinforcement learning with the reward of recognition confidence.

3 Methodology

This paper aims to mitigate the inconsistency problem between bounding boxes and deep representations of text recognition. A reasonable solution is to train the detection module with suitable bounding boxes that can boost the performance of the recognition module. Thus, the issue is how to obtain these appropriate bounding boxes. As illustrated in Fig.2(a), we propose a method with the BoxDQN model structure termed Box Adjuster for adjusting bounding boxes to obtain suitable shapes. BoxDQN accepts an initial bounding box and adjusts it continuously throughout the loop. Then we train the text spotter with adjusted annotated bounding boxes.

The bounding box adjustment is formulated as a sequential decision-making process. In the decision-making process, the agent constantly interacts with the environment and takes a sequence of actions to adjust the bounding box. As shown in Fig.2(b), the agent chooses which action from action space to perform based on the input of four consecutive observations. Following the environment's execution of the selected action, the agent receives the next state and current reward, which can be used to guide the agent's action policy until it achieves a reasonable bounding box by maximising the cumulative rewards. In this section, we first introduce the state, action space, and reward of our model, then describe the components of BoxDQN and its training process. Finally, we detail how our method can be applied to cross-domain problems.

3.1 State and Action Space

Based on the current state and reward, the agent chooses which action to take from action space. So it is crucial to capture abundant information from the state. However, one observation can only provide limited information for the agent, and it is necessary to make full use of historical observations for making decisions. Thus, we choose four serial observations as the state and the current state can be defined as s_t=$\{o_{t-3}, o_{t-2}, o_{t-1}, o_t\}$, where o_t denotes the current observation at step t. A single observation is composed of background, foreground, and box-coordinates, denoted by o_t=$\{background, foreground_t, box\text{-}coordinates_t\}$. The background area is four times the size of the initial bounding box. The $foreground_t$ is cropped from the background by a minimum enclosing rectangle of the bounding box at step t. The $box\text{-}coordinates_t$ represents the coordinates of text in background at step t. We have 16 actions in action space which are combinations of 4 vertexes and 4 directions. As we can see from Fig.2(b), the first action in action spaces implies that the top-left vertex of the quadrangle moves down by one pixel.

3.2 Text Recognition-based Reward

The goal of BoxDQN is to capture appropriate bounding boxes for better recognition. Therefore, a reliable reward is needed to guide the agent to automatically adjust the bounding boxes. We select a few representative text recognition algorithms, including CRNN [37], RARE [38] and others. The average recognition confidence among them is regarded as a reward, so the reward at step t can be formulated as $r_t = conf_{t+1} - conf_t$, where $conf_t = Conf(foreground_t)$, $conf_t$ denotes the recognition confidence at step t.

$$conf_t = \sum_{k=0}^{N_P} conf_k / max(N_G, N_P), \tag{1}$$

where N_P denotes the number of characters in a prediction word and N_G denotes the number of characters in a ground-truth word. The aim of reinforcement

learning is to maximize the cumulative rewards:

$$G_t = \sum_{k=0}^{T} \gamma^k r_{t+k}, \tag{2}$$

where γ denotes the discount factor and $\gamma \in [0,1]$. Ignoring the discount factor, the cumulative reward is equal to $conf_T - conf_0$, where $conf_T$ refers to the recognition confidence of foreground in the terminal state, T means the maximum number of steps and $conf_0$ refers to the recognition confidence of foreground in the initial state. Because $conf_0$ is invariant and only determined by the initial bounding box, maximizing cumulative rewards means maximizing $conf_T$ without γ.

3.3 BoxDQN Model

With the defined action space, state, and reward, the details of BoxDQN are illustrated in Fig.2(b). The agent is composed of a feature fusion module (FFM) and a transformer encoder [42]. It accepts a single state with four observations as input and outputs 16 dimensional vectors, each of which specifies the appropriate action to take. With two deep convolutional neural networks [19] and a transformer encoder, the FFM is proposed to integrate background, foreground, and box-coordinates. During a bounding box adjustment, BoxDQN receives four observations and outputs the corresponding action according to the current state. Every observation needs to be fused by the FFM successively. The feature maps of the background and foreground are extracted from two convolution neural networks, respectively. We concatenate two image feature maps and the position encoding as the input of the transformer in the FFM. After all four observations are passed through the FFM, the transformer in the agent selects an action from the action space based on the concatenation of four fused feature maps. The bounding box moves in response to the selected action, changing both the box-coordinates and the minimum enclosing rectangle of the box-coordinates, and then the next state starts.

3.4 Domain Adaptation

In many cases, due to the absence of labeled real data, we train and test models using synthetic data. However, the domain gap between synthetic and real data degrades performance on real data. To address domain shift problems, we propose a domain-adaptive approach based on our BoxDQN. As shown in Fig.3, our method consists of four steps: (1) refer to the labeled synthetic data domain as the source domain and the unlabeled real data domain as the target domain, (2) train a text spotter with the labeled synthetic data, (3) use the trained text spotter to generate pseudo-labels on real data and adjust the pseudo-labels by employing the BoxDQN mentioned above, (4) finetune the text spotter with the adjusted bounding boxes on real data.

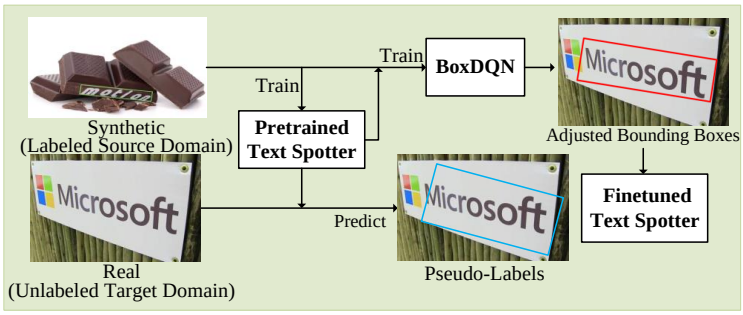

Fig. 3. Illustration of the pipeline with BoxDQN in the area of OCR domain adaptation. We propose a solution that utilizes BoxDQN to tackle domain shift problems.

3.5 Training BoxDQN Model

We use a value-based reinforcement learning method to adjust the bounding boxes, and the training process is presented in Algo.1. During the inner loop of the algorithm, BoxDQN can only adjust one bounding box in a single iteration. Thus, we crop all backgrounds from the source images by the bounding boxes in advance. M is the number of backgrounds. Firstly, the agent selects and executes an action according to an ϵ-greedy policy. The ϵ gradually decreases with iterations from 1.0 to 0.2. Secondly, we present two methods to determine whether the BoxDQN has reached the terminal state. Thirdly, we store a transition $\{s_t, a_t, r_t, s_{t+1}, Term_{t+1}\}$ and sample random mini-batch of transitions in replay memory D. The a_t represents the action at step t and $Term_{t+1}$ represents the terminal state at step t+1, respectively. $Term_{t+1}$ has two types: 0 and 1, where 0 and 1 represent termination and continuance, respectively. Finally, we refer to the training method in the paper[31] that uses a separate network termed $\hat{Q}$ for generating the targets y_j in the Q-learning update. Q represents the BoxDQN agent and has the same network structures as $\hat{Q}$. The $\hat{Q}$-network parameters θ^- are only updated with the Q-network parameters θ every C steps and are held fixed between individual updates. The parameters of Q are updated by optimizing the loss function with stochastic gradient descent. The training loss function is defined as follows:

$$loss = (y_j - Q(s_j, a_j; \theta))^2, \tag{3}$$

where y_j can be formulated as follows:

$$y_j = r_j + (1 - Term_{j+1}) * \gamma * max_{a'} \hat{Q}(a', s_{j+1}; \theta^-). \tag{4}$$

Algorithm 1: Training procedure of the BoxDQN Model

Initialize replay memory D to capacity N
Initialize history memory H to capacity 4
Initialize action-value function Q with random weight θ
Initialize target action-value function $\hat{Q}$ with weight $\theta^- = \theta$
for $episode = 1, M$ **do**
  Initialize observation o_0 according to the initial environment
   $o_0 = \{background, foreground_0, box\text{-}coordinates_0\}$
  Store o_0 in H four times
  Initialize confidence $conf_0 = Conf(foreground_0)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    Otherwise select $a_t = argmax_a Q(s_t, a; \theta)$
    Execute action a_t, observe reward r_t, new
    observation o_{t+1} and new confidence $conf_{t+1}$
    if $conf_{t+1} >= 1.2 * conf_0$ *or t+1==T:* **then**
     $Term_{t+1} = 1$
    else
     $Term_{t+1} = 0$
    end
    Get state s_t from H
    Update H by using o_{t+1}
    Get state s_{t+1} from H
    Store transition $(s_t, a_t, r_t, s_{t+1}, Term_{t+1})$ in D
    Sample random mini-batch of transitions
     $(s_j, a_j, r_j, s_{j+1}, Term_{j+1})$ from D
    Set $s_j = H(j)$ and $s_{j+1} = H(j+1)$
    Set $y_j = r_j + (1 - Term_{j+1}) * \gamma * max_{a'} \hat{Q}(a', s_{j+1}; \theta^-)$
    Perform a gradient descent step on $(y_j - Q(a_j,\ s_j; \theta))^2$ with
     respect to the network paremeters θ
    Every C steps reset $\hat{Q} = Q$
    if $Term_{t+1} == 1$ **then**
     break out
    end
  end
end

4 Experiments

4.1 Datasets

To verify the effectiveness of our method for the end-to-end text spotting methods and the classic two-step methods, we perform experiments on four different datasets. Furthermore, we conduct domain-shift experiments on these datasets to show the robustness of our method in general scenarios.

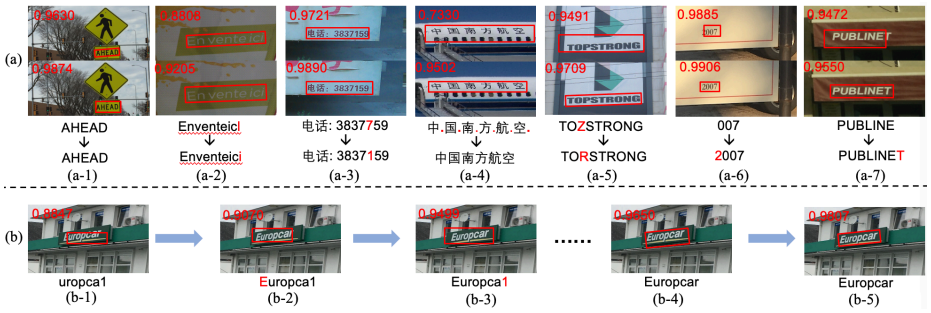

Fig. 4. Qualitative results of BoxDQN. Each pair in (a) is a comparison between the original label (top) and our adjusted bounding boxes (bottom). (a-1) to (a-3) are the results of manual ground truth, and (a-4) to (a-7) are the results on domain-shift, whose bounding boxes are pseudo labels. (b) is a visual display of the adjustment process of BoxDQN. The upper left corner of each image uses red numbers to indicate the recognition confidence.

ICDAR-2013 [17] (IC13) is released during the ICDAR 2013 Robust Reading Competition for focused scene text detection, consisting of high-resolution images, 229 for training and 233 for testing, containing texts in English. The annotations are at word-level using rectangular boxes.
ICDAR-2015 [16] (IC15) is presented for the ICDAR 2015 Robust Reading Competition. All images are annotated with word-level and quadrilateral boxes.
ICDAR-2019ReCTS [44] (ReCTS) is a newly-released large-scale dataset that includes 20,000 training images and 5,000 testing images, covering multiple languages, such as Chinese, English and Arabic numerals. The images in this dataset are mainly collected in the scenes with signboards, All text lines and characters in this dataset are annotated with bounding boxes and transcripts.
ICDAR-2019MLT [32] (MLT19) is a scene text detection dataset, including Chinese, Japanese, English, Arabic, etc. The images in MLT19 are collected from a variety of scenes, and it also contains many real-scene noises.
SynthText-80k [10] & SynthText-MLT [32] are large-scale synthetic datasets, which are adopted as pretraining for our BoxDQN and text spotting models. Furthermore, due to the difference in distribution between the synthetic and real datasets, the synthetic datasets are also used in cross-domain experiments.

4.2 Implementation Details

Baseline. The baseline methods are divided into two distinct categories: (1) two-step methods and (2) end-to-end methods. For two-step methods, due to detected texts are cropped from the image, the detection and recognition are two separate steps. We choose EAST [45] and DBNET [24] to detect the position of the characters, and then CRNN [37] and RARE [38] recognize the content within the

bounding boxes. For end-to-end methods, we adopt FOTS [27], ABCNET [28], and MTS-V3 [22].

Training. A Linux workstation with 32 NVIDIA GeForce 2080Ti (11 GB) is used in our experiments. We train the recognition models in advance on SynthText-80k (for IC13 and IC15) and SynthText-MLT (for MLT19 and ReCTS) as well as on the corresponding real datasets. The trained recognition models are then used in BoxDQN training to continuously adjust and optimize the bounding boxes. The training phase of BoxDQN costs 2 days.

Inference. We evaluate the trained BoxDQN on the training set of the corresponding data and adjust the bounding boxes as the new ground truth to train the detection or spotting models, respectively. The average time taken for BoxDQN to adjust a bounding box is 25ms. In the process of baseline methods evaluation, we follow the official public code repository for training and testing. The datasets in the Section.4 are involved in evaluation.

4.3 Qualitative Results

Our method mainly focuses on adjusting annotated bounding boxes for better text recognition. To verify its effectiveness, we conduct experiments on the four datasets, *i*.e.IC13, IC15, MLT19 and ReCTS. We show the adjustment results of our method on the bounding boxes of different datasets. The qualitative results of bounding boxes in English and Chinese are represented in Fig.4. We find that our bounding boxes can achieve higher credibility in recognition. The (a-1) to (a-3) in Fig.4 show that our BoxDQN can adjust the bounding boxes to make them more suitable for recognition models. It can also correct inaccurate recognition of text in images. Furthermore, we can learn from Fig.4(b) that the adjustment steps of BoxDQN are a step-by-step process. The incorrectly labelled "Europcar" is gradually being correctly recognized. It is worth noting that our recognition confidence is increasing at each step.

4.4 Quantitative Results.

To verify the robustness of our method with those baseline methods, we use the annotated bounding boxes refined by BoxDQN to train the baseline methods. During the quantitative evaluation, the same dataset with the original annotations is also used for training the baseline methods as a comparison. Finally, we test the F-Score metrics of our adjusted bounding boxes under recognition. The gain in Tab.1 indicates the gain after including our BoxDQN.

Two-Step Methods. Two-step methods are those in which the detection model and the recognition model work separately. We choose the combination of [EAST, DBNET] for detection and [CRNN, RARE] for recognition in our experiments. The BoxDQN enhances the bounding boxes, and the detection models are then trained on the adjusted training part of the datasets. The trained models are then evaluated on the test datasets, respectively. From Tab.1, when any combination of two-step pipelines is trained on our BoxDQN refined data, the metrics

Table 1. The quantitative results of our method on the two-step methods. Gain stands for the improvement of the F-Score with and without BoxDQN. We bold the results of each gain to highlight the improvement of the effect by BoxDQN.

Methods	**BoxDQN**	**IC13**		**IC15**		**MLT19**		**ReCTS**	
		F-score	Gain	F-score	Gain	F-score	Gain	F-score	Gain
[EAST+CRNN]	–	84.7	**2.1**	82.2	**1.7**	53.9	**3.0**	69.7	**2.8**
	✓	86.8		83.9		56.9		72.5	
[EAST+RARE]	–	85.5	**1.8**	83.7	**1.9**	55.5	**2.9**	71.1	**2.6**
	✓	87.3		85.4		58.1		73.7	
[DBNET+CRNN]	–	85.2	**1.9**	83.4	**1.7**	55.9	**2.7**	70.0	**2.9**
	✓	87.1		85.1		58.6		72.9	
[DBNET+RARE]	–	85.4	**1.8**	84.7	**1.6**	57.2	**2.4**	73.4	**2.1**
	✓	87.2		86.3		59.6		75.5	

Table 2. The quantitative results of our method on the end-to-end methods. The metrics are the same as the Tab.1.

Methods	**BoxDQN**	**IC13** [17]		**IC15** [16]		**MLT19** [32]		**ReCTS** [44]	
		F-Score	Gain	F-Score	Gain	F-Score	Gain	F-Score	Gain
FOTS [27]	–	83.7	**1.9**	81.5	**1.8**	53.0	**3.1**	70.2	**2.7**
	✓	85.6		83.3		56.1		72.9	
ABCNET [28]	–	86.8	**1.6**	82.4	**1.7**	56.2	**3.3**	72.5	**2.6**
	✓	88.4		84.1		59.5		75.1	
MTS-V3 [22]	–	87.6	**1.5**	83.1	**1.4**	61.2	**2.8**	73.4	**2.3**
	✓	89.1		84.5		64.0		75.7	

obtained are greatly improved. We find that our method has a greater improvement for MLT19 in Tab.1. The gain of [EAST+CRNN] on the IC15 is 1.7%, but for the same pipeline on MLT19, the gain is 3.0%. For more complex OCR scenarios, our method has a more significant improvement. Regardless of any pipeline, the gain of the F-Score can obtain an improvement of at least 1.6%, which is robust to two-step methods.

End-to-End Methods. There are some differences between the end-to-end methods and the two-step methods, mainly in the independence of the detection branch and the recognition branch. We adopt the bounding boxes adjusted by BoxDQN to train the whole end-to-end models rather than the detection models and test the appearance of each metric. Tab.2 shows the results of different end-to-end methods. Our BoxDQN is also helpful for the end-to-end text spotting methods, especially for ABCNET, whose gain is 3.3% and 2.6% on the MLT19 and ReCTS, respectively. The gain of the end-to-end methods is slightly lower compared with the two-step methods. This may be due to the fact that the end-to-end training of text spotters can slightly mitigate inconsistencies in detection and recognition.

Table 3. The experiments results on the cross-domain unlabeled datasets, the annotation information do not used in the datasets. We bold the gain of each pipelines.

Methods	**BoxDQN**	**IC15** [16]		**MLT19** [32]	
		F-Score	Gain	F-Score	Gain
[EAST [45]+CRNN [37]]	–	67.1	**5.4**	41.3	**7.0**
	✓	72.5		48.3	
[EAST [45]+RARE [38]]	–	68.3	**5.3**	42.9	**6.7**
	✓	73.6		49.6	
[DBNET [24]+CRNN [37]]	–	68.3	**5.2**	42.7	**5.4**
	✓	73.5		48.1	
[DBNET [24]+RARE [38]]	–	69.4	**5.3**	44.3	**6.2**
	✓	74.7		50.5	
[FOTS [27]]	–	66.6	**4.8**	39.8	**5.8**
	✓	71.4		45.6	
[ABCNET [28]]	–	69.4	**4.4**	44.4	**5.1**
	✓	73.8		49.5	
[MTS-V3 [22]]	–	71.3	**4.7**	46.7	**4.6**
	✓	76.0		51.3	

4.5 Domain Adaption

Taking into account the domain gap between the synthetic pretraining datasets and the in-the-wild data, we conduct cross-domain experiments to verify the generalization of our method. In detail, we pretrain BoxDQN on the synthetic datasets (SynthText-80k [10] and Synthetic-MLT [32]). After that, we adopt the pre-trained detection models on the relevant real datasets to obtain pseudo bounding boxes. The BoxDQN adjusts the pseudo bounding boxes, and finally the recognition models work on the adjusted bounding boxes to obtain the recognition results. We simulate in-the-wild data through unlabeled IC15 and MLT19, and verify the domain adaptability of our BoxDQN. The results of domain adaption experiments are shown in Tab.3 and Fig.4(a-1,2,3). From Fig.4(a-4,5,6,7), although our method has a slight visual deviation in the adjustment of pseudo-labels, it can improve the confidence and correct the wrong recognition results. Tab.3 proves that our method can improve at least 4.4% in cross-domain datasets.

4.6 Ablation Study

Grid Search. To verify that our BoxDQN is reasonable for adjusting annotated bounding boxes, we compare our method with a grid search policy and include the metric of F-Score in this experiments. In detail, we perform a grid search in each bounding box's four vertices in the directions of up, down, left, and right with a step length of one pixel. The recognition models are trained in advance and give the results with the highest confidence after 10 rounds of grid search as the new ground truth. Refer to Tab.4 for the quantitative results. When

Table 4. Ablation study on grid search. The effect comparison under our BoxDQN and the grid search policy. The experimental dataset is based on IC15 [16].

Methods	Grid Search			BoxDQN		
	Precision	Recall	F-Score	Precision	Recall	F-Score
[EAST [45]+CRNN [37]]	90.3	76.4	82.8	91.0	77.8	**83.9**
[DBNET [24]+RARE [38]]	91.9	80.3	85.7	92.5	80.8	**86.3**
[ABCNET [28]]	93.6	74.5	83.0	94.6	75.7	**84.1**
[MTS-V3 [22]]	93.5	75.2	83.4	94.8	76.2	**84.5**

Table 5. Ablation study on the DQN with only foreground image as input. The effect between our BoxDQN and the original DQN is shown below. The dataset is IC15 [16].

Methods	DQN			BoxDQN		
	Precision	Recall	F-Score	Precision	Recall	F-Score
[EAST [45]+CRNN [37]]	90.5	77.2	83.3	91.0	77.8	**83.9**
[DBNET [24]+RARE [38]]	92.3	79.9	85.7	92.5	80.8	**86.3**
[ABCNET [28]]	94.0	74.5	83.1	94.6	75.7	**84.1**
[MTS-V3 [22]]	94.1	75.5	83.8	94.8	76.2	**84.5**

compared with grid search, BoxDQN can improve recognition accuracy. This qualifies it as an appropriate bounding-box adjustment method in OCR systems and indicates that it does not over-fit the datasets.

Only Foreground Image as Input. Our BoxDQN model adopts a FFM that fuses the foreground, background, and coordinates from the text images. The experimental settings in this section are the same as those in the Sec.4.2. The comparison results are shown in Tab.5, demonstrating that the BoxDQN with more prior information outperforms the classic DQN (86.3 vs. 85.7, [DBNET [24]+RARE [38]] row). And for all of the representative methods, our method has a steady improvement on them. This verifies the robustness of our BoxDQN. More importantly, with the background image as input, our model can handle cases such as those shown in Fig.4(a-7) where the bounding box is slightly shorter than the text transcription.

BoxDQN under Different Iterations. Since our BoxDQN is sensitive to the times of iterations, different iterations have a great impact on the effect. We test the BoxDQN under different iterations and compare the number of iterations corresponding to the best BoxDQN. As shown in Tab.6, the best performance of BoxDQN can be achieved when the number of iterations is set at 20, with minimal resource consumption. This is the same as the number of iterations (20) set in our experiments.

4.7 Exploration on Arbitrarily-shaped Text based on Bezier Curves

To further explore the potential of our approach, we perform experiments on arbitrarily shaped text (TotalText [6] dataset). Our BoxDQN method requires a

Table 6. Ablation study on the number of iterations of BoxDQN. Each row shows the F-Score of the BoxDQN with a different iteration number. We bold the best value of each column.

Iter	[EAST+CRNN]	[DBNET+RARE]	[FOTS]	[ABCNET]	[MST-V3]
5	82.9	85.2	82.3	82.9	83.5
10	83.3	85.8	82.7	83.4	84.0
20	83.9	**86.3**	**83.3**	84.1	84.5
40	**84.0**	86.2	83.1	**84.3**	**84.6**

Table 7. The quantitative results of our method on TotalText. The baseline model is ABCNet with Bezier curves.

Methods	**BoxDQN**	**TotalText** [6]	
		F-Score	Gain
ABCNet [45]	–	61.5	**2.3**
	✓	63.8	

text representation with a fixed number of boundary points for optimization, so we have to convert polygon contour points that do not have a fixed number of points to a representation that does. We can currently only convert arbitrarily-shaped text to a fixed number of control points (8) with the help of Bezier curves. We train our BoxDQN on the SynText150k [28] dataset which contains 150k synthetic arbitrary-shaped text annotated by Bezier curves. Then, we use BoxDQN to adjust the control points of the Bezier curve to obtain the optimal ground truth. The rest of the experimental settings is consistent with the multi-oriented text datasets, except for the differences mentioned above. From Tab.7, we can find there is a considerable improvement(61.5 *vs.* 63.8) in recognition performance when training with the ground-truth Bezier curves optimized by BoxDQN. This experiment illustrates the possibility of extending our approach to arbitrarily-shaped text if there is a more general representation of text boxes.

5 Conclusion and Future Work

In this work, we first analyze the inconsistency between bounding boxes and text recognition, and then present a novel and general preprocessing method called Box Adjuster, which learns the optimal distribution of the text recognition module and delivers it to detection via bounding box adjustment. Our proposed approach is employed exclusively during the training phase, with no additional calculations during the prediction phase. More significantly, the cross-domain problems will be alleviated by utilizing the Box Adjuster. Comprehensive experiments have demonstrated that the proposed approach rationally addresses the aforementioned inconsistency and significantly improves the performance of both two-step and end-to-end text spotting approaches on standard datasets. In the future, we hope to extend our method for arbitrary-shaped text spotting.

6 Acknowledgements

This work was supported by the National Natural Science Foundation of China 61733007.

References

1. Aftabchowdhury, M.M., Deb, K.: Extracting and segmenting container name from container images. International Journal of Computer Applications **74**(19), 18–22 (2013)
2. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proc. CVPR. pp. 9365–9374 (2019)
3. Bartz, C., Yang, H., Meinel, C.: SEE: towards semi-supervised end-to-end scene text recognition. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 6674–6681. AAAI Press (2018), https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16270
4. Busta, M., Neumann, L., Matas, J.: Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In: 2017 IEEE International Conference on Computer Vision (ICCV) (2017)
5. Caicedo, J.C., Lazebnik, S.: Active object localization with deep reinforcement learning. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (December 2015)
6. Ch'ng, C.K., Chan, C.S.: Total-text: A comprehensive dataset for scene text detection and recognition. In: Proc. ICDAR. vol. 1, pp. 935–942 (2017)
7. Dollar, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(8), 1532–1545 (2014)
8. Dvorin, Y., Havosha, U.E.: Method and device for instant translation (Jun 4 2009), uS Patent App. 11/998,931
9. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: IEEE Conference on Computer Vision & Pattern Recognition (2016)
10. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proc. CVPR. pp. 2315–2324 (2016)
11. He, T., Tian, Z., Huang, W., Shen, C., Qiao, Y., Sun, C.: An end-to-end textspotter with explicit alignment and attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5020–5029 (2018)
12. He, Z., Liu, J., Ma, H., Li, P.: A new automatic extraction method of container identity codes. IEEE Transactions on Intelligent Transportation Systems **6**(1), 72–78 (2005)
13. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. International Journal of Computer Vision (2016)
14. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems **28**, 2017–2025 (2015)
15. Jie, Z., Liang, X., Feng, J., Jin, X., Lu, W., Yan, S.: Tree-structured reinforcement learning for sequential object localization. In: Advances in Neural Information Processing Systems. pp. 127–135 (2016)
16. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al.: Icdar 2015 competition on robust reading. In: ICDAR. pp. 1156–1160 (2015)

17. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: Proc. ICDAR. pp. 1484–1493 (2013)
18. Kong, X., Xin, B., Wang, Y., Hua, G.: Collaborative deep reinforcement learning for joint object search. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 25. Curran Associates, Inc. (2012), https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
20. Li, H., Wang, P., Shen, C.: Towards end-to-end text spotting with convolutional recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
21. Li, H., Wang, P., Shen, C., Zhang, G.: Show, attend and read: A simple and strong baseline for irregular text recognition. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 8610–8617 (2019)
22. Liao, M., Pang, G., Huang, J., Hassner, T., Bai, X.: Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. pp. 706–722. Springer (2020)
23. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: Textboxes: A fast text detector with a single deep neural network. In: Thirty-first AAAI conference on artificial intelligence (2017)
24. Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: Proc. AAAI. pp. 11474–11481 (2020)
25. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
26. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
27. Liu, X., Ding, L., Shi, Y., Chen, D., Yan, J.: Fots: Fast oriented text spotting with a unified network. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
28. Liu, Y., Chen, H., Shen, C., He, T., Wang, L.: Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
29. Mathe, S., Pirinen, A., Sminchisescu, C.: Reinforcement learning for visual object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2894–2902 (2016)
30. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. Computer Science (2013)
31. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (Feb 2015), http://dx.doi.org/10.1038/nature14236

32. Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.C., Liu, C.l., et al.: Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1582–1587. IEEE (2019)
33. Peng, X., Huang, Z., Chen, K., Guo, J., Qiu, W.: Rlst: A reinforcement learning approach to scene text detection refinement. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 1521–1528. IEEE (2021)
34. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis & Machine Intelligence **39**(6), 1137–1149 (2017)
35. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)
36. Schroth, G., Hilsenbeck, S., Huitl, R., Schweiger, F., Steinbach, E.G.: Exploiting text-related features for content-based image retrieval. In: 2011 IEEE International Symposium on Multimedia, ISM 2011, Dana Point, CA, USA, December 5-7, 2011 (2011)
37. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence **39**(11), 2298–2304 (2016)
38. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
39. Song, L., Yin, G., Liu, B., Zhang, Y., Yu, N.: Fsft-net: Face transfer video generation with few-shot views. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 3582–3586. IEEE (2021)
40. Tang, J., Zhang, W., Liu, H., Yang, M., Jiang, B., Hu, G., Bai, X.: Few could be better than all: Feature sampling and grouping for scene text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4563–4572 (2022)
41. Tsai, S.S., Chen, H., Chen, D.M., Schroth, G., Girod, B.: Mobile visual search on printed documents using text and low bit-rate features. In: IEEE International Conference on Image Processing (2011)
42. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
43. Wang, H., Huang, S., Jin, L.: Focus on scene text using deep reinforcement learning. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 3759–3765. IEEE (2018)
44. Zhang, R., Zhou, Y., Jiang, Q., Song, Q., Li, N., Zhou, K., Wang, L., Wang, D., Liao, M., Yang, M., et al.: Icdar 2019 robust reading challenge on reading chinese text on signboard. In: 2019 international conference on document analysis and recognition (ICDAR). pp. 1577–1581. IEEE (2019)
45. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: East: an efficient and accurate scene text detector. In: Proc. CVPR. pp. 5551–5560 (2017)

46. Zitnick, C.L., Dollar, P.: Edge boxes: Locating object proposals from edges. In: European Conference on Computer Vision (2014)