

# Dynamic Low-Resolution Distillation for Cost-Efficient End-to-End Text Spotting

Ying Chen<sup>2\*</sup>, Liang Qiao<sup>1,2\*</sup>, Zhanzhan Cheng<sup>2</sup>, Shiliang Pu<sup>2†</sup>,  
Yi Niu<sup>2</sup>, and Xi Li<sup>1†</sup>

<sup>1</sup> Zhejiang University, Hangzhou, China

<sup>2</sup> Hikvision Research Institute, Hangzhou, China

{chenying30,chengzhanzhan,pushiliang.hri,niuyi}@hikvision.com  
{qiaoliang, xilizju}@zju.edu.cn

**Abstract.** End-to-end text spotting has attached great attention recently due to its benefits on global optimization and high maintainability for real applications. However, the input scale has always been a tough trade-off since recognizing a small text instance usually requires enlarging the whole image, which brings high computational costs. In this paper, to address this problem, we propose a novel cost-efficient Dynamic Low-resolution Distillation (DLD) text spotting framework, which aims to infer images in different small but recognizable resolutions and achieve a better balance between accuracy and efficiency. Concretely, we adopt a resolution selector to dynamically decide the input resolutions for different images, which is constrained by both inference accuracy and computational cost. Another sequential knowledge distillation strategy is conducted on the text recognition branch, making the low-res input obtains comparable performance to a high-res image. The proposed method can be optimized end-to-end and adopted in any current text spotting framework to improve the practicability. Extensive experiments on several text spotting benchmarks show that the proposed method vastly improves the usability of low-res models. The code is available at <https://github.com/hikopensource/DAVAR-Lab-OCR/>.

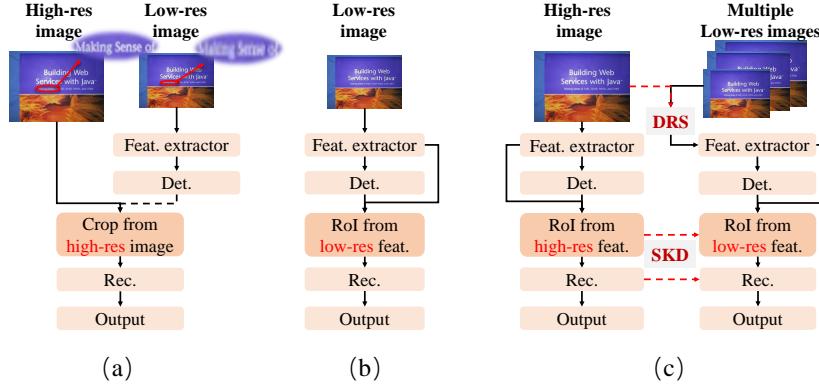
**Keywords:** End-to-End Text Spotting, Dynamic Resolution, Sequential Knowledge Distillation

## 1 Introduction

Research on scene text spotting has achieved great progress and been successfully applied in many fields such as finance, education, transportation, *etc*. The traditional process of text spotting is usually divided into two sub-tasks: text detection [66, 1, 54, 31] and recognition [46, 7, 50, 10]. To reduce the error accumulation between two tasks and the maintenance cost, many works have been proposed in an end-to-end manner [27, 3, 19, 48, 29, 42, 41]. To further improve the real-time performance of the model, some works exquisitely design different geometric

---

\* Authors contributed equally. † Corresponding Authors.



**Fig. 1.** (a) is the offline two-staged text spotter, which can use different resolutions for two tasks while cannot be globally optimized. (b) is the ordinary end-to-end text spotter, where the recognizer can only receive the low-res ROI features map when using the low-res input. (c) is our proposed DLD framework, where the low-res network can dynamically select small but feasible resolutions and reconstruct the high-res features.

representations or lighter network architectures [33,34,35,55,53]. However, most works only report the results based on a fixed and carefully selected input resolution, but whose performances are usually seriously affected by the resolution changing in different situations.

In the traditional pipeline of two-staged text spotting, as shown in Figure 1(a), to save the inference cost, we can firstly detect text from a down-sampled image and then crop text regions from the original high-res image for recognition. It will not damage the overall performance to some extent since the two tasks can be optimized separately. However, once enjoying the benefits such as global optimization and lower maintenance cost brought by end-to-end text spotters, we have to face the resolution choice problem: images can only be resized into a predefined scale. If we want to achieve a higher efficiency using low-res inputs, as illustrated in Figure 1(b), many small texts will lose their discriminative features from the beginning of the network and are hard to be recognized. This makes end-to-end text spotter low practicability in many realistic situations.

This problem is mainly attributed to the different characteristics and resolution sensitivities of two sub-tasks of end-to-end text spotters [6]. When people read the text in a low-res image, they may easily identify whether an object belongs to a text. However, the blurred texture may influence the recognition since it is a more fine-grained sequential classification task. In fact, different texts may have different difficulties for recognition. People can sometimes correctly infer some of the low-res text according to the other recognizable characters and their semantic context meanings [10,62,5]. Nevertheless, when the image keeps down-sampling, it will gradually lose the distinct features for recognition. It means

that for an image containing text with varied sizes and locations, a minimum resolution must exist to make all of the instances recognizable.

Therefore, a better way to balance accuracy and computational cost is to infer different images in different scales [49,60,67]. Moreover, in order to make the network prefer to choose the smaller scales with minimal accuracy drop, we borrow the idea of resolution Knowledge Distillation (KD) [61,26,39], which can enhance the performance of the low-res student using the knowledge transferred from the high-res teacher.

In this paper, we propose a novel framework named *Dynamic Low-resolution Distillation* (DLD) text spotting, which aims to dynamically choose feasible input resolutions for the end-to-end text spotter under a resolution KD schema, as shown in Figure 1(c). Specifically, In DLD, the student network adopts a lightweight *Dynamic Resolution Selector* (DRS) to find a suitable down-sampled resolution to preserve the teacher’s performance. Given a group of candidate down-sampled scales, DRS is optimized to find the best resolution with a minimal performance drop under both accuracy and computational cost supervision. On the other hand, to enhance the recognizability of low-res text, we emphasize the sequence information that the model extracts and integrate it with the *Sequential Knowledge Distillation* (SKD) strategy in the recognition part. The loss of SKD is composed of feature-based L2 loss and the sequence-level beam search output loss, which effectively increases the performance of those *low-res but recognizable* text instances. DLD is a self-consistent framework, where the proposed two tasks can work together organically and achieve effective mutual promotion. The SKD prompts the DRS to choose a relatively lower resolution, and the DRS provides varied ratios of resolution pairs to enhance the SKD to be more robust and focus on the scale-unrelated features.

The major contributions of this paper are as follows: (1) We first study the input resolution problem on end-to-end text spotting tasks and propose a Dynamic Low-resolution Distillation text spotting framework that can effectively enhance the performance and reduce the computational cost. (2) We propose a sequential KD strategy with a dynamic resolution selector that allows the model to choose a small but recognizable input scale. (3) Extensive experiments and ablation studies demonstrate the effectiveness of our method.

## 2 Related Works

### 2.1 End-to-End Text Spotting

Whether to employ Region of Interests (RoI) operations, current end-to-end text spotters can be divided into two types: two-staged and one-staged models.

Two-staged end-to-end text spotters usually involve RoI-like operations to crop detected regions from feature maps for the following recognition task. Methods [28,29,30,37,42,64] usually follow the Faster-RCNN [44]/Mask-RCNN framework [15] to detect text regions and then crop the RoI regions into small features

for recognition using RoI-pooling/RoI-Align operations. To recognize text in arbitrary shapes and improve detection efficiency, some works adopt the segmentation-based methods in the detection stage and then carefully design novel RoI operations to rectify the text into regular shapes, such as RoIRotate [33], RoI-Slide [11], TPS [41,51], BezierAlign [34,35], rectified RoI-Pooling [2], etc.

In one-staged models, text instances are directly decoded from the global feature map without any RoI operation. [58] directly detects characters in different categories using multi-class segmentation. [53] decodes the gathered pixel-level feature vector into sequence with the proposed PG-CTC decoder. In the work of [40], the authors use mask attention to map different instances into different feature map channels and then predict individual text in each channel.

Both two types of methods suffer from the problem of input scales. Compared with two-staged methods that can reshape the RoI features into a uniform size, one-staged methods somehow face more challenges about text scales and usually require a large number of training samples.

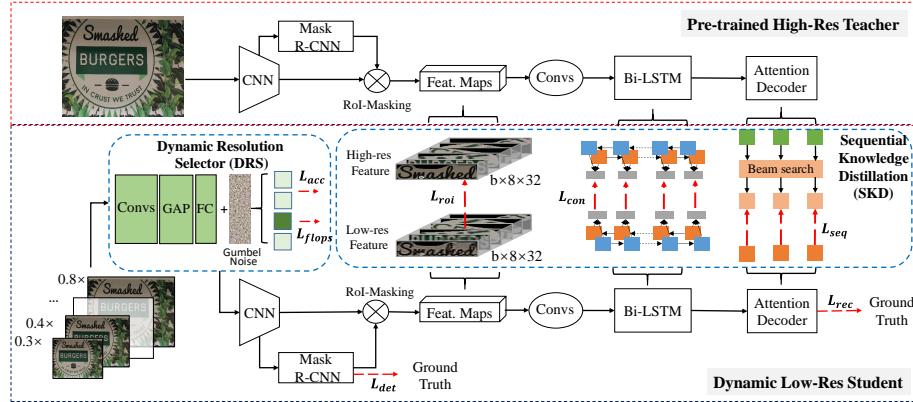
## 2.2 Knowledge Distillation

Knowledge Distillation (KD) [20] was first proposed to transfer the capacity of a large teacher network to a small student. This learning paradigm continues to evolve in the following years and has been applied in many areas, such as image recognition [59], object detection [39], semantic segmentation [18], text recognition [4]. In addition to transferring knowledge between different networks, resolution KD [56,61,26] has also been widely used to train a low-res student with the help of a high-res teacher, which can be well adapted to improve some low-res applications like face recognition [12,13]. [39] firstly studies the low-res KD in object detection tasks and proposes an aligned multi-scale training method to align features in different levels.

However, these methods only conduct resolution KD in a fixed input resolution scale, which somehow limits the generalization ability of the model when objects are displayed in varied scales.

## 2.3 Dynamic Resolution

Since images have different difficulties to recognize, many works are proposed to assign images with dynamic input resolutions [49,60,52,67]. [49] proposes a reinforcement learning approach to dynamically identify when and where to use high-res data conditioned on the corresponding low-res data. [60] uses different sub-networks to cope with samples with different difficulties. [52] learns how to use different scaling strategies for different objects. [67] designs a resolution predictor to choose feasible input resolutions, which can reduce the computational cost while maintain performance.



**Fig. 2.** Illustration of the proposed DLD framework. It contains a fixed pre-trained high-res teacher and a dynamic low-res student that aims to obtain comparable performance. The DRS dynamically selects a small resolution for the student with minimal performance reduction. The SKD process helps the student capture the inter-sequence information and be able to recognize text in low-resolution.

### 3 Methodology

#### 3.1 Overview

As shown in Figure 2, we propose a Dynamic Low-resolution Distillation (DLD) framework. It follows the setting of resolution distillation [56,61,26], where the teacher and student use the same network architecture but were input with images in different resolutions. The high-res teacher network is well pre-trained and then fixed in the following training. The student aims to achieve comparable performance in some lower resolutions. It mainly attributes to two parts: (1) a Dynamic Resolution Selector (DRS) to choose the appropriate resolution for different input images, and (2) a Sequential Knowledge Distillation (SKD) strategy to capture the semantic sequence information and improve the recognizability of the low-res instances. The whole framework is end-to-end trainable.

#### 3.2 Baseline Text Spotting Model

We adopt a Mask-RCNN-based[15] two-staged end-to-end text spotting framework [37,29,42,30] as the base model. The detection branch follows the standard Mask-RCNN implementation, which can predict the mask region with its bounding box for any text shape. For the input image  $I \in R^{H \times W \times 3}$ , the multi-scale features are extracted through the backbone of ResNet-50 [16] and a feature pyramid network (FPN) [32]. For the text recognition task, the recognition features are firstly cropped from the global feature map via the ROI-Masking operation[42] and then uniformly resized into a fixed size  $H_r \times W_r$ .  $H_r$  and  $W_r$

are set to 8 and 32 pixels separately in all experiments. These feature maps go through six convolution layers and are then extracted the sequence information by the bidirectional long short-term memory (Bi-LSTM) [21] module. The final character sequences are decoded by an attention-based decoder [7].

We firstly train a strong high-resolution teacher and then fix it in following distillation training. The teacher network are trained under the supervision from both text detection and recognition annotations as follows,

$$L_{teacher} = L_{det} + L_{rec}, \quad (1)$$

where the detection part contains the losses from bounding boxes regression, classification, and instance segmentation.

### 3.3 Dynamic Resolution Selector

Inspired by [67], we propose a *Dynamic Resolution Selector* (DRS) to help model inference images in more feasible scales. Here, we predefine a group of candidate down-sampled scales for student network, *e.g.*, images can be selected in the range  $[0.8 \times, 0.3 \times]$  of teacher’s resolution. The target of DRS is to find a suitable scale in the group with minimal performance reduction compared to the teacher. The selection criteria are that decreasing the resolution should be rewarded while performance decline would be penalized. Specifically, The DRS is a lightweight residual network composed of 10 Convolution layers (Convs), a Global Average Pooling (GAP) layer, and a Fully Connected (FC) layer. Given a high-res image  $I$  and  $k$  candidate down-sampled scale factors  $\{s_1, s_2, \dots, s_k\}$ , the DRS first predicts the probability vector  $p=[p_1, p_2, \dots, p_k]$  by the network and then transforms  $p$  into binary decisions  $h=[h_1, h_2, \dots, h_k] \in \{0, 1\}^k$  indicating which scale factor to select. To optimize DRS, the network first does forward calculations based on the given pre-trained teacher with the high-res input  $I$  and all of the corresponding low-res images  $\{I_{s_1}, I_{s_2}, \dots, I_{s_k}\}$ . We use  $y$  and  $\{y_{s_1}, y_{s_2}, \dots, y_{s_k}\}$  to denote the predicted probability distribution of teacher and students, respectively. The loss generated in terms of accuracy can be formed as:

$$L_{acc} = KL(\sum_{i=1}^k h_i y_{s_i}, y), \quad (2)$$

where  $KL(\cdot)$  is the KL divergence for the recognition results. The recognition feature maps are cropped according to the detection ground truth (GT) during training to ensure the number of recognition results is consistent. The optimization target is to make one of the  $h_i$  be 1, and the others are 0. The selected scale with the teacher’s nearest recognition results and the highest detection accuracy will obtain the minimum loss.

To prevent the DRS module from converging to the maximum scale and encourage it to choose a smaller image as much as possible, we directly penalize it with its forward computation cost as follows:

$$L_{flops} = \sum_{i=1}^k h_i T_i, \quad (3)$$

where  $T_i$  is the forward FLOPS under input  $I_{s_i}$ . Since images containing different instances have different FLOPS, we use the pre-computed average value.

The overall supervision of DRS is as follows:

$$L_{DRS} = L_{acc} + \gamma L_{flops}, \quad (4)$$

where  $\gamma$  is the parameter to balance the weight between accuracy and computational cost.

**Gumbel Softmax Trick.** Notice that the process of converting  $p$  into one-hot  $h$  is non-differentiable. Here, we adopt the Gumbel-Softmax sampling trick [22]. Specifically, we first add Gumbel noise  $g_j$  to the discrete random variable  $p_j$ , and then draw discrete samples from the above distribution as:

$$h_i = \begin{cases} 1, & i = \arg \max_j (\log(p_j) + g_j) \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

where  $g_j = -\log(-\log(u_j))$  is calculated based on the i.i.d samples  $u_j$ , where  $u_j \sim Uniform(0, 1)$ . In the above procedure, the arg max operation can be approximated by softmax operation as follows:

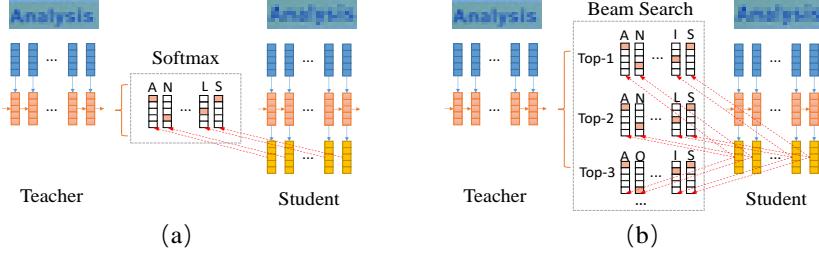
$$h_i = \frac{\exp((\log(p_i) + g_i)/\tau)}{\sum_{j=1}^k \exp((\log(p_j) + g_j)/\tau)}, \quad (6)$$

where  $\tau$  is the temperature parameter. During training, using a lower  $\tau$  can make the expectation of sampling closer to the result of arg max, but the gradient variance will be large. Adopting a higher  $\tau$  can make the gradient variance smaller, but the expectation of sampling will be close to the average distribution. Here, we initialize with a larger  $\tau$  at the beginning and gradually decrease it as  $\tau = \sigma^{epoch} \tau_{init}$ , where  $\tau_{init}$  is the initial temperature and  $\sigma$  is a decay factor.

### 3.4 Sequential Knowledge Distillation

Text recognition is a sequential classification problem where the sequence information is vital for capturing the semantic meaning [46,10,62]. For example, although some characters are easily confused in low-res, such as ‘i’ and ‘l’ in Figure 3, people can still recognize them in a word. This inspires us to dig out further the model’s deeper potential, making DRS choose the smaller scale as much as possible. Therefore, we propose a *Sequential Knowledge Distillation* (SKD) strategy to help the low-res network extract semantic information under the supervision of its teacher. Here, we only explore the optimization problem of the text recognition task since it is more likely to be the bottleneck of the overall performance of the current end-to-end text spotting framework, which will also be demonstrated in the following experiments.

Specifically, given a high-res input  $I$  and the selected low-res image  $I_s$ , we use  $\mathcal{F}^{roi}, \mathcal{F}_s^{roi} \in \mathbb{R}^{H \times W \times C}$  to separately represent the cropped recognition features of



**Fig. 3.** The comparison of (a) logit-based Knowledge Distillation and (b) sequence-level Knowledge Distillation in sequence decoding.

teacher and student, where  $H$ ,  $W$ , and  $C$  denote the feature map’s height, width, and channel, respectively. These features are cropped using ROI-Masking with the detection GT and resized into a uniform shape. After a stack of convolutions and a Bi-LSTM module, the contextual information can be further extracted, which are represented as  $\mathcal{F}^{con}, \mathcal{F}_s^{con} \in \mathbb{R}^{N \times C}$ , where  $N$  is the length of the hidden state. In these two stages of the network, we adopt the feature-based KD strategy to set up L2 loss as:

$$L_{roi} = \frac{1}{HWC} \sum_{i=1}^H \sum_{j=1}^W \sum_{c=1}^C \|\mathcal{F}^{roi}[i, j, c] - \mathcal{F}_s^{roi}[i, j, c]\|_2, \quad (7)$$

$$L_{con} = \frac{1}{NC} \sum_{i=1}^N \sum_{c=1}^C \|\mathcal{F}^{con}[i, c] - \mathcal{F}_s^{con}[i, c]\|_2. \quad (8)$$

In the final decoding stage, different from the distillation process that aggregates logit-based loss [4] over the sequence, we borrow the idea of the sequence-level knowledge distillation [25] to better capture sequence information, as shown in Figure 3. The student is trained based on the output from the top- $k$  beam search [43] results of the teacher network. It helps the student retain the context information in a sequence as much as possible. To ensure the optimization speed, we only choose the results with the top-3 scores. Specifically, given the input  $\mathcal{F}^{con}$  and the attention-based decoder, we use  $p(q|\mathcal{F}^{con})$  to represent the predicted sequence distribution over all possible sequences  $q \in \mathcal{Q}$ . Then, the sequence-level knowledge distillation can be formulated and then approximated as follows:

$$\begin{aligned} L_{seq} &= - \sum_{q \in \mathcal{Q}} p(q|\mathcal{F}^{con}) \log p(q|\mathcal{F}_s^{con}) \\ &= - \sum_{q \in \mathcal{Q}} \mathbb{I}\{q = \hat{y}_1, \hat{y}_2, \hat{y}_3, \dots\} \log p(q|\mathcal{F}_s^{con}), \\ &\approx - \log \sum_{k=1}^K p(q = \hat{y}_k|\mathcal{F}_s^{con}) \end{aligned} \quad (9)$$

where  $\hat{y}_k$  is the result with the top- $k$  beam search score of the teacher model. Here, we simply set  $K=1$  to save training time.

Finally, the overall SKD is optimized as:

$$L_{SKD} = L_{roi} + \eta_1 L_{con} + \eta_2 L_{seq}, \quad (10)$$

where  $\eta_1$  and  $\eta_2$  are the hyper-parameters to balance the magnitude of  $L_{roi}$ ,  $L_{con}$  and  $L_{seq}$ .

### 3.5 Optimization

The proposed DLD framework is optimized to recognize images in both high accuracy and efficiency with end-to-end training. The overall loss of the student network is generated from three parts: the original text detection and recognition loss, the loss from DRS to balance accuracy and computational cost, and the loss from SKD to improve the representational capacity of the low-res model.

$$L = L_{det} + \lambda_1 L_{rec} + \lambda_2 L_{DRS} + \lambda_3 L_{SKD}, \quad (11)$$

where  $\lambda_1, \lambda_2$  and  $\lambda_3$  are weight balancing parameters.

In every epoch of the training stage, the student network will do forward calculations for  $k$  times for all candidate resolutions. The backward loss of  $L_{DRS}$  will only be propagated in the lightweight DRS module. The loss of  $L_{det}$ ,  $L_{rec}$ , and  $L_{SKD}$  will only be conducted on the branch with the maximum  $h_i$  and will be not propagated to the DRS.

## 4 Experiments

### 4.1 Implementation Details

**Datasets.** We list the datasets used in this paper as follows. We evaluate our method on three popular text spotting benchmarks: (1) *ICDAR2013* [24] (*IC13*) that only contains horizontal text, (2) *ICDAR2015* [23] (*IC15*) that includes oriented text, and (3) *Total-Text* [8] (*TT*) that involves many curved text. For the teacher network, we firstly pre-train it on *SynthText-800K* [14] and then fine-tune with a mixture dataset which includes 7k images filtered from *ICDAR-MLT2017* [38] and all training images in *IC13*, *IC15*, and *TT*. In the following KD training stage, the teacher will be fixed, and the student network can be initialized using the teacher's weights.

**Experiment Settings.** The base model's architecture is described in Section 3.2. Teacher and student models share the same training settings. All models are trained by the AdamW [36] optimizer with batch-size=3. The KD training lasts for 50 epochs and uses an initial learning rate of  $1 \times 10^{-3}$ . The learning rate is divided by 10 at the 30-th epoch and the 40-th epoch. The parameter  $\tau_{init}$  is set as 5 and decay factor  $\sigma=0.965$ . The Bi-LSTM module has 256 hidden units. For weight balancing parameters, we set  $\gamma=0.1$  and others as  $\eta_1=\eta_2=\lambda_1=\lambda_2=\lambda_3=1$ .

According to the scales of text instances in different datasets, we choose the basic inference resolutions for the teacher, *i.e.*, ‘S-768’ for IC13, ‘S-1280’ for IC15, and ‘S-896’ for Total-Text, where the prefix of ‘S-’ represents the input images are resized by a fixed shorter side.

To obtain a strong baseline of the teacher network, we conduct widely-used data augmentation strategies as follows: (1) instance aware random cropping, (2) randomly scaling the shorter side of the input images to lengths in the range scales  $[0.3\times, 1.0\times]$  of the basic resolutions, (3) random rotation with angle randomly chosen from  $[-15^\circ, +15^\circ]$ , (4) applying random brightness, jitters, and contrast to input images. In both training and testing stages, the student’s DRS resolution ratios range is set as  $\{0.8, 0.7, 0.6, 0.5, 0.4, 0.3\}$ . All experiments are implemented in Pytorch with  $8\times 32$  GB-Tesla-V100 GPUs under CUDA-10.0 and CUDNN-7.6.3.

## 4.2 Results on Text Spotting Benchmarks

The effectiveness of the proposed DLD are compared with other three settings: (1) *Vanilla Multi-scale*: a single model that is trained in multiple scales and tested in a fixed scale. (2) *DRS-only*: with the proposed DRS, the student network removes the supervision from the distillation. (3) *SKD-only*: with the proposed SKD, the student keeps using the 1/2 scale compared with the teacher.

Table 1 shows the experimental results on three benchmarks. The result of *Vanilla Multi-scale* with high-res input can be treated as the original upper bound of the model. If the inputs are in low-resolutions, although the FLOPS can be optimized by approximately 75%, the accuracies will be dramatically decreased, *e.g.*, the *End-to-End* results of *General/None* are decreased by 8.5% (82.9% *vs.* 74.4%) in *IC13*, 6.6% (69.5% *vs.* 62.9%) in *IC15* and 6.9% (62.3% *vs.* 55.4%) in *Total-Text*, respectively. Using the SKD to transfer knowledge from the high-res teacher into the low-res student, in the result of *SKD-only*, we can see that the performances on low-res can be effectively increased by 4.4%/4.2%/4.2% in three datasets compared with *Vanilla Multiscale*, respectively.

Equipped with the DRS module, in *DRS-only*, we are able to tune the tendencies of the model to balance the accuracy and computational cost by different  $\gamma$ . When the model additionally integrates with SKD, under the entire *DLD* framework, where the student network can select more low-res scales without performance drop, the overall performances are further optimized. Specifically, when we set  $\gamma=0.1$ , the model can achieve the comparable or even better accuracy (82.7% *vs.* 82.9%, 70.9% *vs.* 69.5%, 63.9% *vs.* 62.3%) than that of high-res input, and with about 50% FLOPS costs. Suppose we want the model more tend to be cost-efficient and set  $\gamma=0.3$ . The models’ FLOPS can be reduced to a similar level as that all using 1/2-resolution inputs, and the accuracies can be further improved by 2.8%/2.2%/2.3% compared with *SKD-only*, respectively. More statistical and visualized analysis are in the supplementary materials.

We conduct the following ablation experiments in Section 4.3, 4.4, 4.5 on *Total-Text* [8] and use ‘None’ to denote lexicon-free End-to-End result and ‘Full’ to represent the result based on the lexicon combined all images.

**Table 1.** Results on three text spotting benchmarks. ‘S’, ‘W’ and ‘G’ separately mean recognition with strong, weak and generic lexicon [23]. ‘Full’ indicates lexicons of all images are combined, and ‘None’ means lexicon-free [8]. ‘H’ and ‘L’ in column ‘Type’ indicates whether the inference is carried out with high- or low-resolution input. FLOPS is the average *floating point operations*.

Dataset	Training Method	Type	Input Size	End-to-End (%)					Word Spotting (%)					FLOPS
				S	W	G	None	Full	S	W	G	None	Full	
IC13	Vanilla Multi-Scale	H	S-768	86.9	86.6	82.9	-	-	91.4	91.0	86.3	-	-	142.9G
		L	S-384	80.9	78.9	74.4	-	-	85.2	82.7	77.3	-	-	35.8G
	SKD-only	L	S-384	84.1	82.8	78.8	-	-	88.0	86.5	81.7	-	-	35.8G
	DRS-only ( $\gamma=0.1$ )	L	Dynamic	85.7	84.8	80.7	-	-	90.1	88.9	84.0	-	-	80.7G
	DRS-only ( $\gamma=0.3$ )	L	Dynamic	83.7	82.0	77.6	-	-	87.8	85.8	80.5	-	-	48.8G
	DLD ( $\gamma=0.1$ )	L	Dynamic	86.5	85.7	82.7	-	-	90.9	89.9	86.1	-	-	71.5G
IC15	DLD ( $\gamma=0.3$ )	L	Dynamic	85.6	84.4	81.6	-	-	90.0	88.6	84.9	-	-	41.6G
	Vanilla Multi-Scale	H	S-1280	78.0	74.4	69.5	-	-	81.4	77.2	71.7	-	-	517.2G
		L	S-640	72.2	67.8	62.9	-	-	75.7	70.8	65.3	-	-	129.3G
	SKD-only	L	S-640	75.4	71.7	67.1	-	-	78.9	74.6	69.6	-	-	129.3G
	DRS-only ( $\gamma=0.1$ )	L	Dynamic	76.2	72.1	66.8	-	-	79.8	75.2	69.3	-	-	298.8G
	DRS-only ( $\gamma=0.3$ )	L	Dynamic	73.6	68.9	63.7	-	-	76.4	71.5	66.3	-	-	163.6G
TT	DLD ( $\gamma=0.1$ )	L	Dynamic	79.0	75.7	70.9	-	-	82.4	78.6	73.3	-	-	261.8G
	DLD ( $\gamma=0.3$ )	L	Dynamic	78.1	73.5	69.3	-	-	81.1	76.4	71.2	-	-	148.3G
	Vanilla Multi-Scale	H	S-896	-	-	-	62.3	71.4	-	-	-	65.2	75.9	206.7G
		L	S-448	-	-	-	55.4	66.5	-	-	-	58.1	71.1	52.0G
	SKD-only	L	S-448	-	-	-	59.6	68.9	-	-	-	62.6	73.5	52.0G
	DRS-only ( $\gamma=0.1$ )	L	Dynamic	-	-	-	60.9	70.4	-	-	-	63.5	75.0	119.2G
	DRS-only ( $\gamma=0.3$ )	L	Dynamic	-	-	-	58.8	68.9	-	-	-	61.6	73.6	75.0G
	DLD ( $\gamma=0.1$ )	L	Dynamic	-	-	-	63.9	73.7	-	-	-	66.4	77.8	103.0G
	DLD ( $\gamma=0.3$ )	L	Dynamic	-	-	-	61.9	71.9	-	-	-	64.0	75.9	62.1G

### 4.3 Ablation Studies on Sequential Knowledge Distillation

**Different Distillation Losses.** SKD contains losses from three parts: the RoI feature’s loss  $L_{roi}$ , the contexture feature’s loss  $L_{con}$ , and the beam search output’s loss  $L_{seq}$ . Here, we conduct different experiments based on the model of *SKD-only* to evaluate the importance degrees of these losses, and the result is shown in Table 2. It is easy to know that without any KD loss, the model will fall into the *Vanilla Multi-Scale* setting. From the result, we can see that  $L_{seq}$  has the greatest impact on distillation, which surpasses the result without distillation by 3.0% on ‘None’ and 1.4% on ‘Full’. By combining all three losses, the model achieves 4.2%/2.4% improvements.

**Different Knowledge Distillation Settings.** We also conduct experiments on different KD strategies. Based on the setting of Resolution KD framework without DRS (teacher with S-896, and student with S-448), we compared our model with other two works: (1) Bhunia *et al.* [4]: contains four types of KD losses (Logits’ Distillation, Character Localised Hint, Attention Distillation, Affinity Distillation) that are designed for text recognition, and (2) Qi *et al.* [39]: a KD strategy designed for the detection stage. Table 3 shows the experimental results. For the recognition KD, the results demonstrate that the performance of our proposed SKD surpasses [4] by 0.4%/0.6%. This is mainly because of the effectiveness of the sequence-level distillation strategy in some low-res texts.

**Table 2.** Ablation on different distillation losses for SKD.

Training Method	$L_{roi}$	$L_{con}$	$L_{seq}$	End-to-End(%)	
				None	Full
SKD-only (S-448)	✓			55.4	66.5
		✓		56.9	66.9
			✓	57.5	67.2
		✓	✓	58.4	67.9
	✓	✓	✓	59.6	68.9

**Table 3.** Ablation on different knowledge distillation settings.

Distillation Method	End-to-End (%)	
	None	Full
SKD	59.6	68.9
Bhunia <i>et al.</i> [4]	59.2	68.3
SKD replaced Logits [4]	58.7	67.5
Qi <i>et al.</i> [39]	55.8	67.0
SKD+Qi <i>et al.</i> [39]	59.8	69.2

**Table 4.** Ablation on different RoI scales. ‘†’ means model has extra deconv modules.

Training Method	Type	Teacher RoI Scale	Student RoI Scale	End-to-End (%)		FLOPS
				None	Full	
Vanilla Multi-Scale	H	(8×32)	-	62.3	71.4	206.7G
	H	(16×64)	-	63.1	72.1	227.3G
DLD	L	(8×32)	(8×32)	63.9	73.7	103.0G
	L	(16×64)	(16×64)	64.8	74.1	112.3G
	L	(16×64)	(8×32)†	64.5	73.3	119.2G

We simply replace the sequence-level distillation in SKD with the logits-based distillation adopted in [4], and we can see the performance will drop by 0.9% on ‘None’ and 1.4% on ‘Full’, respectively. For the detection KD, the results show that the enhancement of integration with detection distillation is limited. This is because, in the current model, the detection performance in low-resolution has only a small gap between that of high-resolution.

**Different RoI Scales.** Recognition feature scale is a factor that affects performance under the current Mask-RCNN-based framework. Here, we conduct experiments to evaluate its influence, whose results are illustrated in Table 4. The results show that larger RoI scales would help models obtain higher performance but inevitably bring extra computational cost. On the other hand, if teacher and student use different RoI scales, the model cannot directly perform distillation. So, we add a deconvolutional (deconv) module [63] to align the smaller student’s features with the larger teacher’s features. This also simulates a feature-level super-resolution process. From the results, we can see that compared with directly using a larger student RoI scale, conducting super-resolution will even degrade the performance and increase the FLOPS.

#### 4.4 Ablation on Dynamic Resolution Selector

**Different Candidate Scales.** The set of candidate student scales is usually defined by experience. In Table 5, we compared the results under different candidate sets. The sets containing a single value are the same as the *SKD-only* model. We can see that the group of smaller candidates {0.5,0.4,0.3} obtains lower inference costs and accuracy than that of the larger group {0.8,0.7,0.6}. With more candidates values, the model can be optimized to find a better balance between accuracy and computational cost. Nevertheless, the training cost will be somehow increased.

**Table 5.** Ablation on different candidate scales. The training time reports the average time used to train the model for an epoch.

Training Method	Resolution Scales	End-to-End (%)		FLOPS	Training time (min/epoch)
		None	Full		
DLD	{0.5}	59.6	68.9	52.0G	7.8
	{0.8, 0.7, 0.6}	63.6	73.4	128.8G	9.0
	{0.5, 0.4, 0.3}	58.9	68.0	45.9G	8.4
	{0.8, 0.7, 0.6, 0.5, 0.4, 0.3}	63.9	73.7	103.0G	10.2

**Table 6.** Ablation on parameter  $\gamma$ .**Table 7.** Ablation on parameter  $\tau_{init}$ .

Training Method	$\gamma$	End-to-End (%)		FLOPS
		None	Full	
DLD	0.1	63.9	73.7	103.0G
	0.2	63.2	72.4	82.8G
	0.3	61.9	71.9	62.1G
	0.4	59.2	69.5	50.6G
	0.5	56.0	66.4	38.4G

Training Method	$\tau_{init}$	End-to-End (%)		FLOPS
		None	Full	
DLD	1	63.6	71.9	115.8G
	3	63.5	73.2	110.2G
	5	63.9	73.7	103.0G
	7	63.8	72.6	96.7G
	9	63.2	72.5	98.5G

**Balance between Accuracy and Computational Cost.**  $\gamma$  is a vital parameter to control the DRS’s tendencies about accuracy and efficiency. Table 6 shows how the changes of  $\gamma$  influence the model. As the increasing of  $\gamma$ , the model tends to choose more minor input scales and achieves more efficient computational cost. However, the accuracy will be somehow reduced. This parameter can be controlled flexibly and provide a straightforward guide on the resolution choice under different requirements.

**Different Temperature Parameter.** Table 7 shows how  $\tau_{init}$  influence the student’s performance. This parameter affects the results of the convergence of the DRS module to some extent. It can be easily tuned once  $\gamma$  be fixed.

#### 4.5 Studies on Different End-to-End Text Spotters

To demonstrate the effectiveness of our method, besides the basic Mask-RCNN-based text spotter, we also conducted experiments on the other two different frameworks based on the open-sourced code: (1) Text Perceptron [41], a two-staged text spotter whose text detection branch is segmentation-based, and (2) MANGO [40], a one-staged text spotter. Since MANGO has no explicit detection branch and directly recognizes text globally, we only report *End-to-End* result based on Intersection over Union(IoU)=0.1 constraint as reported in [40]. Other results are reported based on IoU=0.5. We calculate the *Text Recognition (Rec)* accuracy using the GT of detection in inference.

The experimental results are displayed in Table 8. In the *Vanilla Multi-Scale* setting, as the input scales decreases, it is not hard to understand that almost all of the accuracies will be declined. Nevertheless, in the Mask-RCNN-based framework, the performance drops in *Rec* and *End-to-End* are faster than that of *Det*. It means the text detection task can still work well in some relatively low resolutions, while the text recognition task becomes the overall bottleneck of End-to-End performance. This also proves the motivation we mentioned initially. In contrast, for the other two text spotters, the *Det* performance degrades faster

**Table 8.** The ablations experiments on scale changes in *Vanilla Multi-Scale* and the compared result when adopting DLD on different text spotting frameworks. ‘*Det*’ is the Hmean metric of text detection task. ‘*Rec*’ is the Accuracy metric of text recognition task. ‘*E2E*’ stands for ‘End-to-End’. ‘Full’ indicates lexicons of all images are combined and ‘None’ means lexicon-free. FPS is the average *frames per second*.

Training Method	Input Size	Mask-RCNN-based				Text Perceptron [41]				MANGO [40]		
		Det (%)	Rec (%)	E2E(%)	FPS	Det (%)	Rec (%)	E2E(%)	FPS	E2E(%)	FPS	
Vanilla Multi-Scale	S-896	85.3	73.6	62.3	71.4	7.9	85.3	73.1	66.0	74.6	9.5	66.2
	S-768	85.7	72.8	61.2	71.1	8.6	85.1	72.6	64.7	74.2	11.4	67.1
	S-640	86.1	71.7	60.5	70.2	9.0	85.2	72.2	64.3	73.1	13.3	66.5
	S-512	84.9	66.9	58.7	69.3	9.4	82.5	68.9	61.3	70.5	14.8	61.9
	S-384	82.3	58.6	52.0	63.5	9.8	77.4	60.7	54.1	64.3	16.5	50.9
	S-256	76.5	42.5	38.3	48.5	12.0	62.2	45.2	40.5	50.2	17.8	29.2
DLD ( $\gamma=0.1$ )	Dynamic	85.8	74.8	63.9	73.7	9.1	85.6	75.3	67.1	76.4	13.2	67.8
DLD ( $\gamma=0.3$ )	Dynamic	85.1	73.1	61.9	71.9	9.7	81.7	73.5	63.6	72.8	15.8	62.5
										73.8	10.4	

when compressing the input size, which also jointly influences the End-to-End performance. This is because, in Mask-RCNN-based text spotters, there are a lot of preset anchors to capture different scales of text, but the segmentation task is relatively more sensitive to scales [45].

When we adopt the proposed DLD on these text spotters, we can see that all the models with  $\gamma=0.1$  can obtain even higher accuracies and faster speed than that of high-res result in *Vanilla Multi-Scale*. Although the inference speed can be further accelerated with  $\gamma=0.3$ , the *End-to-End* performance drops on Text Perceptron and MANGO are much larger than that on Mask-RCNN-based. Since DLD does not involve distillation loss in the text detection task, it is predictable that the *End-to-End* performance can be further improved with the help of the text detection knowledge transfer. Recall the results reported in Table 1, and we can find that the FLOPS is negatively correlated to the FPS but is not in equal proportion, because many operations are optimized to calculate in parallel and different platforms might behave differently. Nevertheless, reducing the computational cost is also vital for the low-end devices.

## 5 Conclusion

This paper proposes a novel *Dynamic Low-resolution Distillation* (DLD) framework for cost-efficient end-to-end text spotting, aiming to recognize images in different *low but recognizable* resolutions. The model integrates a dynamic low-resolution selector that can choose different down-sampled scales. A sequential knowledge distillation strategy is then adopted to make the model be able to recognize images with lower resolutions and hence achieve a better resolution-performance balance. Experiments show that our method can effectively enhance the practicability of end-to-end text spotters in many complicated situations.

## References

1. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: CVPR. pp. 9365–9374 (2019) 1
2. Baek, Y., Shin, S., Baek, J., Park, S., Lee, J., Nam, D., Lee, H.: Character region attention for text spotting. In: ECCV. vol. 12374, pp. 504–521 (2020) 4
3. Bartz, C., Yang, H., Meinel, C.: See: towards semi-supervised end-to-end scene text recognition. In: AAAI. pp. 6674–6681 (2018) 1
4. Bhunia, A.K., Sain, A., Chowdhury, P.N., Song, Y.Z.: Text is text, no matter what: Unifying text recognition using knowledge distillation. In: ICCV. pp. 983–992 (2021) 4, 8, 11, 12
5. Bhunia, A.K., Sain, A., Kumar, A., Ghose, S., Chowdhury, P.N., Song, Y.Z.: Joint visual semantic reasoning: Multi-stage decoder for text recognition. In: ICCV. pp. 14940–14949 (2021) 2
6. Chen, X., Jin, L., Zhu, Y., Luo, C., Wang, T.: Text recognition in the wild: A survey. ACM Comput. Surv. **54**(2), 42:1–42:35 (2021) 2, 23
7. Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., Zhou, S.: Focusing Attention: Towards Accurate Text Recognition in Natural Images. In: ICCV. pp. 5076–5084 (2017) 1, 6
8. Ch'ng, C.K., Chan, C.S.: Total-text: A Comprehensive Dataset for Scene Text Detection and Recognition. In: ICDAR. vol. 1, pp. 935–942 (2017) 9, 10, 11
9. Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., Wang, H.: PP-OCR: A practical ultra lightweight OCR system. CoRR **abs/2009.09941** (2020) 23
10. Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In: CVPR. pp. 7098–7107 (2021) 1, 2, 7
11. Feng, W., He, W., Yin, F., Zhang, X., Liu, C.: Textdragon: An end-to-end framework for arbitrary shaped text spotting. In: ICCV. pp. 9075–9084 (2019) 4
12. Ge, S., Zhao, S., Li, C., Li, J.: Low-resolution face recognition in the wild via selective knowledge distillation. IEEE TIP **28**(4), 2051–2062 (2019) 4
13. Ge, S., Zhao, S., Li, C., Zhang, Y., Li, J.: Efficient low-resolution face recognition via bridge distillation. IEEE TIP **29**, 6898–6908 (2020) 4
14. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic Data for Text Localisation in Natural Images. In: CVPR. pp. 2315–2324 (2016) 9
15. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: ICCV. pp. 2980–2988 (2017) 3, 5
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR. pp. 770–778 (2016) 5
17. He, L., Gao, F., Hou, W., Hao, L.: Objective image quality assessment: a survey. Int. J. Comput. Math. **91**(11), 2374–2388 (2014) 25
18. He, T., Shen, C., Tian, Z., Gong, D., Sun, C., Yan, Y.: Knowledge adaptation for efficient semantic segmentation. In: CVPR. pp. 578–587 (2019) 4
19. He, T., Tian, Z., Huang, W., Shen, C., Qiao, Y., Sun, C.: An End-to-End TextSpotter with Explicit Alignment and Attention. In: CVPR. pp. 5020–5029 (2018) 1
20. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR **abs/1503.02531** (2015) 4
21. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997) 6

22. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: ICLR (2017) 7
23. 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) 9, 11
24. 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: ICDAR. pp. 1484–1493 (2013) 9
25. Kim, Y., Rush, A.M.: Sequence-level knowledge distillation. In: Su, J., Carreras, X., Duh, K. (eds.) EMNLP. pp. 1317–1327 (2016) 8
26. Li, D., Yao, A., Chen, Q.: Learning to learn parameterized classification networks for scalable input images. In: ECCV. vol. 12374, pp. 19–35 (2020) 3, 4, 5
27. Li, H., Wang, P., Shen, C.: Towards End-to-end Text Spotting with Convolutional Recurrent Neural Networks. In: ICCV. pp. 5248–5256 (2017) 1
28. Li, H., Wang, P., Shen, C.: Towards end-to-end text spotting with convolutional recurrent neural networks. In: ICCV. pp. 5248–5256 (2017) 3, 22
29. Liao, M., Lyu, P., He, M., Yao, C., Wu, W., Bai, X.: Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. IEEE TPAMI 1(1) (2019) 1, 3, 5
30. Liao, M., Pang, G., Huang, J., Hassner, T., Bai, X.: Mask TextSpotter v3: Segmentation Proposal Network for Robust Scene Text Spotting. In: ECCV (2020) 3, 5
31. Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: AAAI. pp. 11474–11481 (2020) 1
32. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature Pyramid Networks for Object Detection. In: CVPR. pp. 2117–2125 (2017) 5
33. Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., Yan, J.: FOTS: Fast Oriented Text Spotting with a Unified Network. In: CVPR. pp. 5676–5685 (2018) 2, 4
34. Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L.: ABCNet: Real-time Scene Text Spotting with Adaptive Bezier-Curve Network. In: CVPR. pp. 9809–9818 (2020) 2, 4
35. Liu, Y., Shen, C., Jin, L., He, T., Chen, P., Liu, C., Chen, H.: Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. IEEE TPAMI (2021) 2, 4
36. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019) 9
37. Lyu, P., Liao, M., Yao, C., Wu, W., Bai, X.: Mask Textspotter: An End-to-End Trainable Neural Network for Spotting Text with Arbitrary Shapes. In: ECCV. pp. 71–88 (2018) 3, 5
38. Nayef, N., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., Luo, Z., Pal, U., Rigaud, C., Chazalon, J., Khelif, W., Luqman, M.M., Burie, J., Liu, C., Ogier, J.: ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification - RRC-MLT. In: ICDAR. pp. 1454–1459 (2017) 9
39. Qi, L., Kuen, J., Gu, J., Lin, Z., Wang, Y., Chen, Y., Li, Y., Jia, J.: Multi-scale aligned distillation for low-resolution detection. In: CVPR. pp. 14443–14453 (2021) 3, 4, 11, 12
40. Qiao, L., Chen, Y., Cheng, Z., Xu, Y., Niu, Y., Pu, S., Wu, F.: MANGO: A mask attention guided one-stage scene text spotter. In: AAAI. pp. 2467–2476 (2021) 4, 13, 14, 26

41. Qiao, L., Tang, S., Cheng, Z., Xu, Y., Niu, Y., Pu, S., Wu, F.: Text perceptron: Towards end-to-end arbitrary-shaped text spotting. In: AAAI. pp. 11899–11907 (2020) 1, 4, 13, 14
42. Qin, S., Bissacco, A., Raptis, M., Fujii, Y., Xiao, Y.: Towards unconstrained end-to-end text spotting. In: ICCV. pp. 4704–4714 (2019) 1, 3, 5
43. Rabiner, L., R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989) 8
44. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: NeurIPS. pp. 91–99 (2015) 3
45. Richardson, E., Azar, Y., Avioz, O., Geron, N., Ronen, T., Avraham, Z., Shapiro, S.: It's all about the scale - efficient text detection using adaptive scaling. In: WACV. pp. 1833–1842 (2020) 14
46. 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 TPAMI **39**(11), 2298–2304 (2017) 1, 7
47. Smith, R.: An overview of the tesseract OCR engine. In: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23–26 September, Curitiba, Paraná, Brazil. pp. 629–633. IEEE Computer Society (2007) 23
48. Sun, Y., Zhang, C., Huang, Z., Liu, J., Han, J., Ding, E.: Textnet: Irregular text reading from images with an end-to-end trainable network. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) ACCV. pp. 83–99 (2018) 1
49. Uzkent, B., Ermon, S.: Learning when and where to zoom with deep reinforcement learning. In: CVPR. pp. 12342–12351 (2020) 3, 4
50. Wan, Z., He, M., Chen, H., Bai, X., Yao, C.: Textscanner: Reading characters in order for robust scene text recognition. In: AAAI. pp. 12120–12127 (2020) 1
51. Wang, H., Lu, P., Zhang, H., Yang, M., Bai, X., Xu, Y., He, M., Wang, Y., Liu, W.: All you need is boundary: Toward arbitrary-shaped text spotting. In: AAAI. pp. 12160–12167 (2020) 4
52. Wang, H., Kembhavi, A., Farhadi, A., Yuille, A.L., Rastegari, M.: ELASTIC: improving cnns with dynamic scaling policies. In: CVPR. pp. 2258–2267 (2019) 4
53. Wang, P., Zhang, C., Qi, F., Liu, S., Zhang, X., Lyu, P., Han, J., Liu, J., Ding, E., Shi, G.: Pgnet: Real-time arbitrarily-shaped text spotting with point gathering network. In: AAAI. pp. 2782–2790 (2021) 2, 4
54. Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S.: Shape robust text detection with progressive scale expansion network. In: CVPR. pp. 9336–9345 (June 2019) 1
55. Wang, W., Xie, E., Li, X., Liu, X., Liang, D., Zhibo, Y., Lu, T., Shen, C.: Pan++: towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. IEEE TPAMI (2021) 2
56. Wang, Y., Sun, F., Li, D., Yao, A.: Resolution switchable networks for runtime efficient image recognition. In: ECCV. vol. 12360, pp. 533–549 (2020) 4, 5
57. Wick, C., Reul, C., Puppe, F.: Calamari - A high-performance tensorflow-based deep learning package for optical character recognition. CoRR **abs/1807.02004** (2018) 23
58. Xing, L., Tian, Z., Huang, W., Scott, M.R.: Convolutional character networks. In: ICCV. pp. 9126–9136 (2019) 4
59. Xu, K., Rui, L., Li, Y., Gu, L.: Feature normalized knowledge distillation for image classification. In: ECCV. vol. 12370, pp. 664–680 (2020) 4
60. Yang, L., Han, Y., Chen, X., Song, S., Dai, J., Huang, G.: Resolution adaptive networks for efficient inference. In: CVPR. pp. 2366–2375 (2020) 3, 4

61. Yang, T., Zhu, S., Chen, C., Yan, S., Zhang, M., Willis, A.R.: Mutualnet: Adaptive convnet via mutual learning from network width and resolution. In: ECCV. vol. 12346, pp. 299–315 (2020) 3, 4, 5
62. Yu, D., Li, X., Zhang, C., Liu, T., Han, J., Liu, J., Ding, E.: Towards accurate scene text recognition with semantic reasoning networks. In: CVPR. pp. 12110–12119 (2020) 2, 7
63. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: CVPR. pp. 2528–2535 (2010) 12
64. Zhang, P., Xu, Y., Cheng, Z., Pu, S., Lu, J., Qiao, L., Niu, Y., Wu, F.: TRIE: end-to-end text reading and information extraction for document understanding. In: ACM MM. pp. 1413–1422 (2020) 3
65. Zhong, H., Tang, J., Wang, W., Yang, Z., Yao, C., Lu, T.: ARTS: eliminating inconsistency between text detection and recognition with auto-rectification text spotter. CoRR **abs/2110.10405** (2021) 23, 26
66. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: EAST: An Efficient and Accurate Scene Text Detector. In: CVPR. pp. 2642–2651 (2017) 1
67. Zhu, M., Han, K., Wu, E., Zhang, Q., Nie, Y., Lan, Z., Wang, Y.: Dynamic resolution network. In: NeurIPS (2021) 3, 4, 6

## A Appendices

### A.1 More Analysis

In this section, we will provide more statistical analysis and show some visualization results to demonstrate the effectiveness of the proposed method.

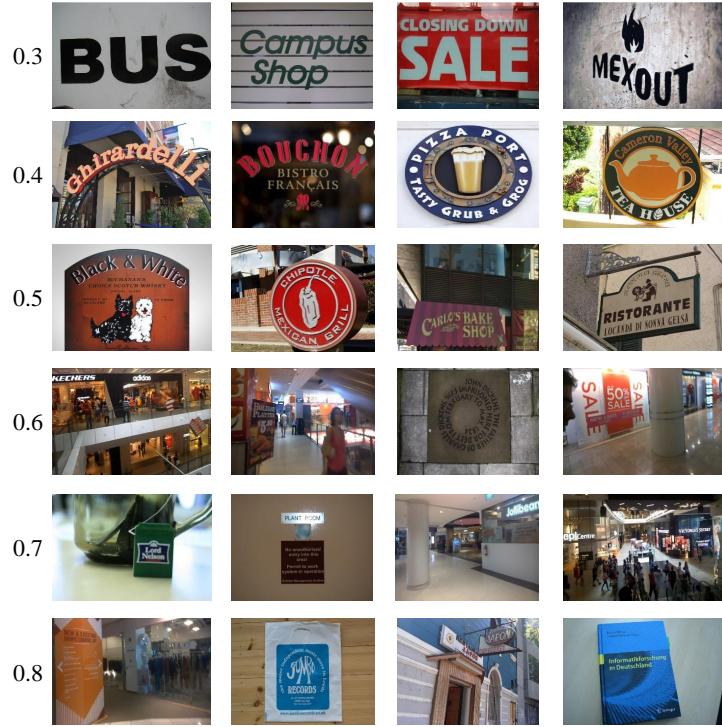
**Selected Scales Distribution.** We counted the distributions of the input scales selected by DRS under the entire DLD framework, as illustrated in Table A1. As we can see, the model dynamically chooses different input scales for different images. When  $\gamma=0.1$ , the model is more inclined to accuracy, and more selections are concentrated on the scales of 0.6, 0.7, or 0.8. With the  $\gamma$  increases, the model will select more low-res scales. The scales distributions are also slightly different for different datasets. For example, IC15 chooses more large scales than IC13 since it contains a larger proportion of small text.

**Table A1.** The distributions of the selected scales using different  $\gamma$  in DLD.

Scales	$\gamma = 0.1$			$\gamma = 0.3$			$\gamma = 0.5$		
	IC13	IC15	TT	IC13	IC15	TT	IC13	IC15	TT
0.3	0.4%	0.2%	0%	4.3%	1.6%	1.6%	17.5%	11.4%	15.0%
0.4	2.6%	0.6%	6.7%	15.9%	14.6%	11.3%	36.0%	35.2%	35.3%
0.5	7.7%	6.2%	7.3%	46.5%	44.2%	47.7%	42.3%	44.4%	42.7%
0.6	24.1%	21.0%	26.3%	21.1%	29.2%	28.7%	3.8%	8.6%	7.0%
0.7	34.8%	39.3%	38.3%	12.0%	10.0%	10.7%	0.4%	0.4%	0%
0.8	30.4%	32.7%	27.4%	0.4%	0.4%	0%	0%	0%	0%

**Visualization of the DRS.** Dynamic Resolution Selector (DRS) aims to assign different images with different input scales dynamically. Under the candidate scales set  $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ , we randomly select some of the images from three benchmarks in different scales and visualize them in Figure A1. On the overall trend, we can see that DRS is optimized to choose smaller input scales for images containing large text and bigger input scales for images with small text. Here, the term ‘large’ or ‘small’ reflects the relative ratio of the text to the whole image instead of the instance’s absolute resolution. The scale that benefits the overall performance would be selected for images containing both large and small text, such as the sample in row-3 and column-1.

**Visualization of the SKD.** The proposed Sequential Knowledge Distillation (SKD) strategy could help low-resolution images obtain similar performance to those in high-resolution. In Figure A2, we demonstrate some end-to-end results before and after adopting the SKD strategy under the 1/2 resolution inputs (S-384 for IC13, S-640 for IC15 S-448 for TT), respectively. We can see that with the proposed SKD, the model can correctly recognize some of the confused text through context semantic information.



**Fig. A1.** The selection results of *DLD-DRS-only* for different scales in IC13, IC15, TT.

**Visualization of the DLD.** Combining with SKD, the model makes the smaller scales have more chances of selecting the DRS module. As for those low-res inputs, DRS can help them select different feasible scales in a cost-efficient way. We illustrate some of the comparisons of End-to-End results between the model of *DLD-SKD-only* and the entire DLD framework in Figure A3. The model of *DLD-SKD-only* infers images in the 1/2 scales. We can see that although some of the blurred text has been correctly recognized, some text still failed to predict since the infeasible fixed scale, such as the first group of images as shown. Under the entire DLD framework, images can be resized to a suitable scale, achieving better performance with minimal computational cost changes. For those easily recognized images such as the fourth group images in Figure A3, the DLD allows them to choose a lower input scale without performance reduction.

We select some of the samples that are correctly recognized in high-res but failed in low-res by *Vanilla Multi-Scale* model, as visualized in Figure A4. We also provide the compared results predicted by our proposed DLD model. As

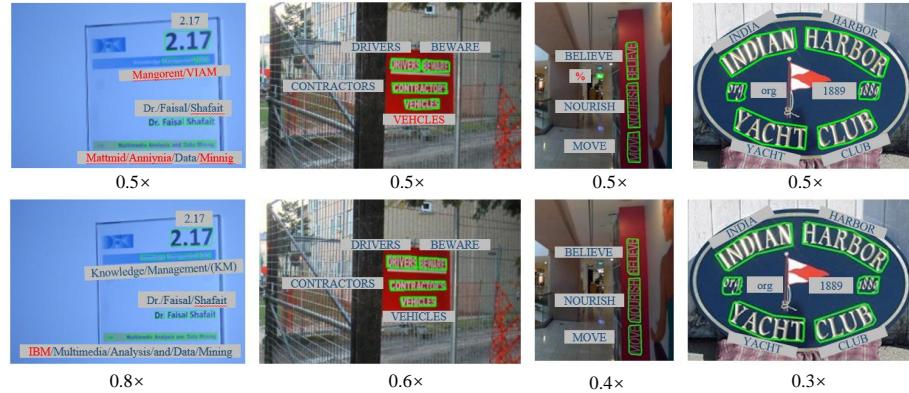


**Fig. A2.** The visualization of end-to-end results in IC13, IC15, and TT under 1/2 resolution inputs. The first row shows the results without using SKD (*Vanilla Multi-Scale*) and the second row adopts SKD (*DLD-SKD-only*). Text in red are incorrectly recognized.

we can see, under DLD framework, the model has dynamically selected different scales and mostly correctly recognized these instances in the low resolution.

**Feature Distribution of SKD.** The target of the SKD is to narrow the feature distribution difference between high resolution and low-resolution inputs. To demonstrate the effectiveness, we visualize the feature distribution in two network positions obtained by Principal Component Analysis (PCA), *i.e.*, (1) *RoI Feature*: the feature map after RoI-Masking operation, and (2) *Contexture Feature*: the feature map after Bi-LSTM module. We compare the feature distribution between high-res and low-res inputs for different models, and the visualization results on three datasets are shown in Figure A5 and Figure A6. As we can see, without using the SKD strategy (*Vanilla Multi-Scale*), the feature differences between high-res and low-res inputs are relatively large. When the model integrates with the SKD (*DLD-SKD-only*), the feature distributions are narrowed. This demonstrates that, although the Multi-Scale training strategy improves the models' robustness on different input scales, the model actually assigns different scales with different feature distributions and results in performance discrepancies when input scales change. The SKD strategy essentially aligns the feature distribution differences and makes the model more robust in low-resolution.

Table A2 records the quantitative differences calculated by L2 distance. Compared with *Vanilla Multi-Scale* training, distillation training can reduce the L2



**Fig. A3.** The visualization of end-to-end results in IC13, IC15, and TT. The first row shows the results under 1/2 input scales in *DLD-SKD-only* and the second row corresponds to the entire *DLD*. The numbers below the images are the down-sampled scales compared to the original high-resolution. Text in red are incorrectly recognized.

	Produkt	TOAST	DESSERT	BEACH
MS	H: Produkt	H: TOAST	H: DESSERT	H: BEACH
DLD	L(0.5x): Produkt	L(0.5x): TOKST	L(0.5x): BERGERS	L(0.5x): BEACE

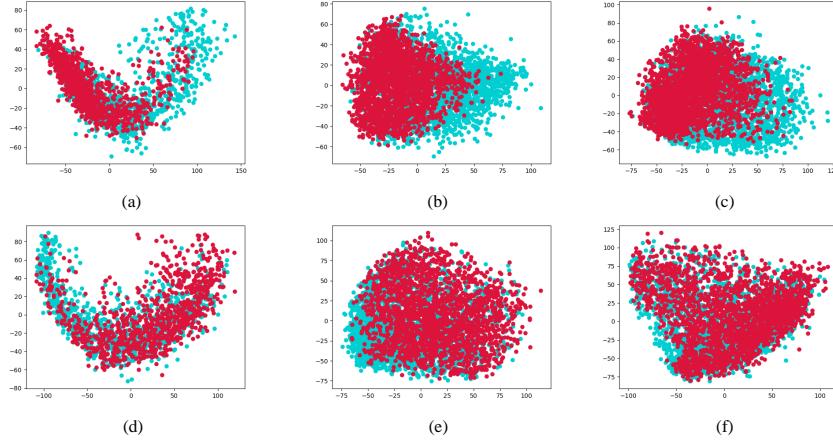
	CHANDRA	Family	Hougang	Ave
MS	H: CHANDRA	H: Family	H: Hougang	H: Ave
DLD	L(0.5x): GLASSI	L(0.5x): Fandy	L(0.5x): Haugang	L(0.5x): Are

**Fig. A4.** The compared recognition results between *Vanilla Multi-Scale* and *DLD*. Characters in red are incorrectly recognized.

distance of *RoI Feature* distribution by 0.13 on IC13, 0.11 on IC15 and 0.13 on Total-Text, respectively. As for the *Contextual Feature*, the distances can be decreased by 0.08, 0.08, 0.10, respectively. Comparing the feature discrepancies in two positions, the RoI feature's differences between high-res and low-res inputs are much larger than the other. This demonstrates that it is unwise only to adopt distillation in the deepest layer. Distillation on RoI features can help the model recover the information in the early phases and prevent over-fitting.

## A.2 Discussion

The technique of End-to-End text spotting has been studying for several years since [28] was proposed. Many advances have demonstrated the advantages of End-to-End models compared to the traditional two-staged pipeline, for example, (1) better performance since freeing from error accumulation between two



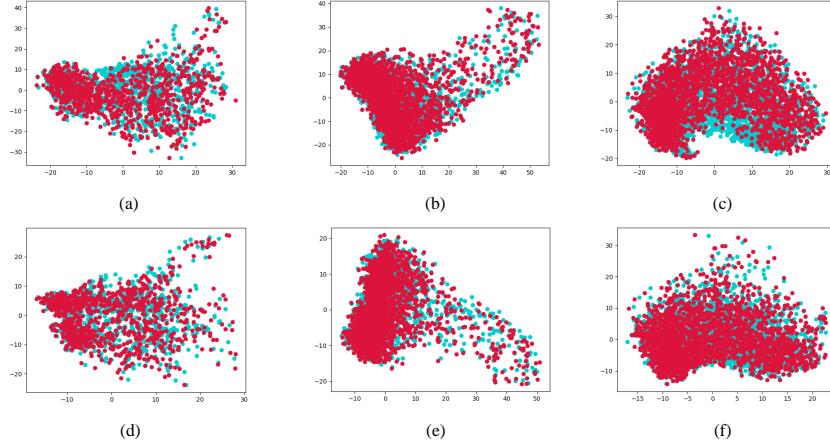
**Fig. A5.** The PCA visualization of the *RoI Feature* distributions on high-res and low-res inputs. (a) *Vanilla Multi-Scale* on IC13. (b) *Vanilla Multi-Scale* on IC15. (c) *Vanilla Multi-Scale* on TT. (d) *DLD-SKD-only* on IC13. (e) *DLD-SKD-only* on IC15. (f) *DLD-SKD-only* on TT. Blue points denote the high resolution distribution and Red points denote the low resolution distribution.

tasks, (2) faster inference and smaller storage requirements by information sharing and jointly optimization (3) lower maintenance cost [6].

However, most of the current Optical Character Recognition (OCR) engines still solve the real tasks in a two-staged way, such as Tesseract [47], PP-OCR [9], Calamari [57], etc. There might exist many reasons, but one of the immediate factors is that an end-to-end model with good performance and robustness is hard to obtain in many real complicated situations. Due to the different characteristics of the sub-tasks, it is not easy to balance two tasks [65]. The text detection task predicts instances' scopes and locations and focuses more on the different scales of coarse-grained features like boundaries. The recognition task is a sequential classification problem that usually requires images on a uniform scale, and it is more concentrated on fine-grained features like textures. Therefore, the input resolution becomes a sensitive factor to the second staged task and may cause the overall performance fluctuation.

To alleviate such problems, besides the intuitive way that makes the model be trained on more samples and with abundant means of augmentations, the proposed DLD framework can effectively reduce the resolution sensitivity of the text recognition task from the feature level. In real applications, the DLD can be flexibly used to enhance the robustness of the end-to-end model.

**Failure cases.** In this section, we demonstrate some failure cases in the testing phase. The failure cases are mainly caused by wrong resolution selection



**Fig. A6.** The PCA visualization of the *Contexture Feature* distributions on high-res and low-res inputs. (a) *Vanilla Multi-Scale* on IC13. (b) *Vanilla Multi-Scale* on IC15. (c) *Vanilla Multi-Scale* on TT. (d) *DLD-SKD-only* on IC13. (e) *DLD-SKD-only* on IC15. (f) *DLD-SKD-only* on TT. Blue points denote the high resolution distribution and Red points denote the low resolution distribution.

and errors inherent in the detection model, here we only focus on the failure cases caused by the wrong resolution selection to reveal the limitations of our proposed framework. As shown in A7, we can see that it is difficult for the resolution selector to choose a suitable resolution when the text size distribution in the images varies greatly. Moreover, there are also a few failure cases assigning large resolutions to images with large text or assigning small resolutions to images with small text, which will bring extra time consumption or makes the text unrecognizable. Nonetheless, our proposed resolution selector works well for most text images and reduces the input resolution without affecting the performance. To demonstrate the effectiveness, as shown in Table A3, we illustrate the distribution of failure cases. We can see from the table that most of the failure cases are caused by errors inherent in the detection model. Moreover, errors due to the introduction of our proposed *DLD* framework only account for 2.51% in IC13, 2.16% in IC15 and 1.94% in TT, respectively. Furthermore, equipped with our proposed sequential distillation, we can effectively correct some of incorrect recognition results.

**Limitations.** Although the proposed DLD framework effectively optimizes the model's performance of accuracy and computational cost, there are still some problems to be solved in the future.

First, the basic high-resolution and the set of candidate down-sampled scales need to be decided manually, which requires sufficient data analysis and certain

Dataset	Training Method	L2 distance of RoI Feature	L2 distance of Contexture Feature
IC13	Vanilla Multi-Scale	0.43	0.20
	DLD-SKD-only	0.30	0.12
IC15	Vanilla Multi-Scale	0.28	0.18
	DLD-SKD-only	0.17	0.10
TT	Vanilla Multi-Scale	0.34	0.19
	DLD-SKD-only	0.21	0.09

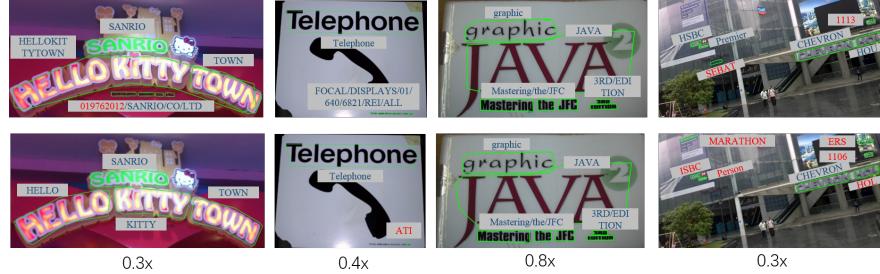
**Table A2.** L2 distances of feature distribution between high-res and low-res inputs on different benchmarks.

Dataset	Vanilla Multi-Scale	DLD	ratio (%)
IC13	✓	✓	78.08
	✓	✗	2.51
	✗	✓	3.49
	✗	✗	15.92
IC15	✓	✓	62.93
	✓	✗	2.16
	✗	✓	4.72
	✗	✗	30.19
TT	✓	✓	64.79
	✓	✗	1.94
	✗	✓	4.74
	✗	✗	28.53

**Table A3.** The distribution of failure cases in IC13, IC15 and TT. 4 means correct text recognition, 7 means incorrect text recognition.

experiences of people. Given an empirical setting, although the model can perform better than simply using multi-scale training and fixed-scale testing, but can hardly find out the theoretically best solution. Enlarging the candidate set's capacity is a way to find more optimal results. However, the training cost will inevitably increase since the selector needs to calculate the forward results of all candidate scales during training. A possible alternative way to achieve dynamic resolution selecting is to predict resolutions in soft labels rather than hard labels, such as quality assessment [17]. However, we still face many challenges since it is hard to define the quality of an image containing different text.

Second, Although the KD strategy can increase the performance of the low-res model, there is always a limit to the competence of the model. When the input image is down-sampled to a small scale that even humans cannot recognize, the KD learning might lead the model to an over-fitting state. Furthermore, the current model only considers the KD problem on the recognition task. However, the detection task is also sensitive to the resolution for the text spotters that adopt the segmentation-based detection branch. Although we have demonstrated the effectiveness to adopt our method on those models, the performances still have large improvement spaces. More experiments should be conducted on the different types of models in the future.



**Fig. A7.** Visualization of some failure cases caused by wrong resolution selection in IC13, IC15, and TT. The first row shows the results under original input scales with *Vanilla Multi-Scale* and the second row corresponds to the entire *DLD*. The numbers below the images are the down-sampled scales compared to the original high-resolution. Text in red are incorrectly recognized.

The last problem is a common problem for most current text spotters. The target of text spotting is to obtain the final text sequences prediction. However, the text detection branch is mostly optimized to obtain high-IoU with the detection Ground Truth (GT), which is not always consistent with text recognition [40,65]. This problem would introduce the new question about how to balance these two tasks. In our DLD, in addition to the GT-oriented optimization, many other balance parameters still need to be set manually, which might also influence the model's performance to some extent.