ODM: A Text-Image Further Alignment Pre-training Approach for Scene Text Detection and Spotting

Chen Duan* duanchen02@meituan.com
Pei Fu* fupei@meituan.com
Shan Guo † guoshan05@meituan.com
Qianyi Jiang jiangqianyi02@meituan.com
Xiaoming Wei weixiaoming@meituan.com

Abstract

*In recent years, text-image joint pre-training techniques have shown promising results in various tasks. However, in Optical Character Recognition (OCR) tasks, aligning text instances with their corresponding text regions in images poses a challenge, as it requires effective alignment between text and OCR-Text (referring to the text in images as OCR-Text to distinguish from the text in natural language) rather than a holistic understanding of the overall image content. In this paper, we propose a new pre-training method called **O**CR-Text **D**estylization **M**odeling (ODM) that transfers diverse styles of text found in images to a uniform style based on the text prompt. With ODM, we achieve better alignment between text and OCR-Text and enable pre-trained models to adapt to the complex and diverse styles of scene text detection and spotting tasks. Additionally, we have designed a new labeling generation method specifically for ODM and combined it with our proposed Text-Controller module to address the challenge of annotation costs in OCR tasks, allowing a larger amount of unlabeled data to participate in pre-training. Extensive experiments on multiple public datasets demonstrate that our method significantly improves performance and outperforms current pre-training methods in scene text detection and spotting tasks. Code is available at ODM.*

1. Introduction

Optical Character Recognition (OCR) has garnered significant attention in the field of computer vision for its remarkable performance in automated data entry, document analysis, instant translation, and more. Most existing methods for obtaining OCR results involve a two-stage process, including a text detection model and a text recognition model, or directly utilizing an end-to-end text spotting model. Furthermore, many of these methods initialize the model through pre-training on the ImageNet dataset [37].

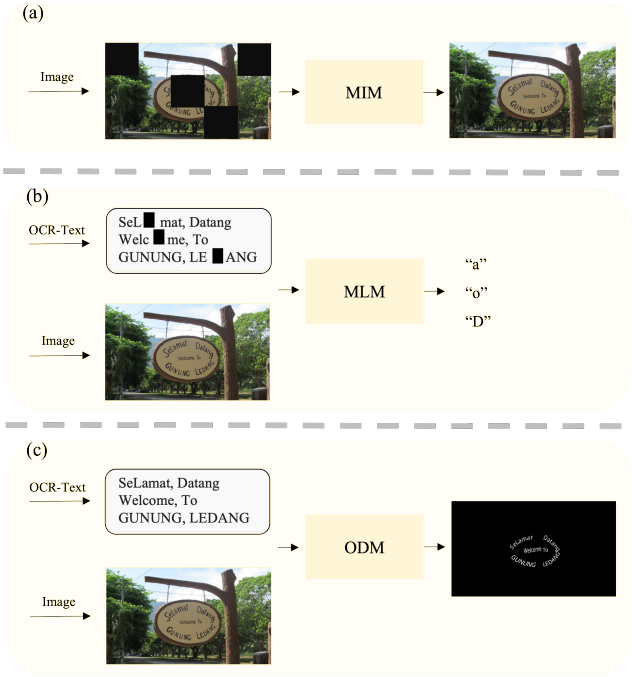

Figure 1. Comparisons of different pre-training strategies. (a) Obtain the pre-trained model through mask image modeling, taking only image embeddings as inputs. (b) Obtain the pre-trained model through mask language modeling, which simultaneously takes both OCR-Text and image as inputs. (c) Our approach obtains the pre-trained model through OCR-Text destylization modeling.

Pre-training techniques have recently gained significant attention for their outstanding performance across a wide range of computer vision tasks. Two commonly used pre-training methods in computer vision are: (1) Masked Image Modeling (MIM) pre-training [1, 3, 10, 49], which

*First Author and Second Author contribute equally to this work.
†Corresponding author

focuses on learning visual contextualized representations solely from images. This method is typically applied to vision-dominated tasks, such as image classification. (2) Masked Language Modeling (MLM) [6, 51, 52], which utilizes both text and image as inputs and extracts semantic information from both modalities. However, when applying these pre-training methods in the OCR field, two specific issues arise: (1) With the MIM-based method, there are instances where the text in the image is completely obscured by the masked patch, primarily due to the relatively small proportion of the text. This has the potential to hinder the pre-trained model's ability to effectively learn textual feature information. (2) The MLM-based method, although achieving weakly supervised training by masking the text input, does not explicitly exploit text location information during training. This can lead to ineffective alignment between text and image features, as well as inadequate handling of image information. These challenges highlight the need for innovative approaches that specifically address the unique requirements of OCR.

In this paper, we propose a novel pre-training technique called OCR-Text Destylization Model (ODM) to address the challenges of text-image alignment in OCR tasks. As illustrated in Fig. 1, unlike existing MIM and MLM methods, ODM introduces a new pixel-level image reconstruction modeling based on text prompts. Since OCR tasks primarily focus on the text within the image while considering other pixels irrelevant, ODM aims to reconstruct a binary image that removes the text style and enforces alignment between the text and OCR-Text. This is achieved by utilizing a pixel-level reconstruction approach instead of the traditional three-channel reconstruction. To further enhance the model's understanding of the text, we propose a Text-Controller module. This module guides the image encoder to identify and interpret the OCR-Text, facilitating the alignment between the text and OCR-Text. Additionally, we have designed a novel method for generating ODM labels, effectively addressing the issue of inadequate pixel-level labels in the dataset. By leveraging font files, text, and location labels, we generate binary images with a unified font style, as illustrated in Fig. 2, which showcases some examples of OCR-Text destylization images. These advancements in ODM and the Text-Controller module, along with the novel label generation method, contribute to improved text-image alignment in OCR tasks.

In summary, the main contributions are three-fold:

(1) We propose a simple yet effective pre-training method called ODM, which focuses on learning features specifically for OCR-Text. By using pixel-level labels with a uniform style, we successfully destylize OCR-Text, improving text comprehension. This crucial feature information enables the pre-trained model to adapt well to various scenarios in text detection and spotting tasks.

(a) (b) (c) (d)

Figure 2. The upper row and lower row represent the original images and their corresponding destylized labels, respectively. (a), (b), (c), and (d) are taken from the ICDAR15 [15], CTW1500 [26], TotalText [5], and LSVT [41] datasets, respectively.

(2) We introduce a novel Text-Controller module that helps regulate the model's output, enhancing its understanding of OCR-Text. With this module, our method does not require a perfect match between the input image and the text pair. As a result, we can utilize weakly annotated data (i.e., using other OCR recognition engines to obtain the text and location in the image and filter it based on recognition confidence, text size, etc.), which can greatly reduce the annotation cost.

(3) Experimental results on public datasets demonstrate that ODM delivers outstanding performance and surpasses existing pre-training techniques across a range of scene text detection and spotting datasets.

2. Related Work

Scene Text Detection. Scene text detectors based on deep learning can primarily be categorized into regression based [11, 18, 32, 38, 42, 57] and segmentation based [22, 23, 46, 47, 50, 55, 58] methods. Regression-based methods perceive scene text as an object and directly regress the bounding box of the text instance. Segmentation-based methods treat the text detection task as a semantic segmentation problem. They obtain a segmentation map by directly segmenting the text instance and subsequently group the segments into a box through post-processing.

Scene Text Spotting. Scene text spotting represents the unification of detection and recognition processes within a singular framework. Numerous studies have improved performance by simultaneously learning detectors and recognizers. In [2, 7, 12, 19, 20, 25], end-to-end implementation is achieved by training the detection and recognition separately. The Mask TextSpotter [21, 30, 31] series performs character segmentation during the recognition process. The ABCNet [27, 28] series obtains detection coordinates through control points of the Bezier curve. SwinTextSpotter [13] and ESTextSpotter [14] use separate detection-recognition heads corresponding to different an-

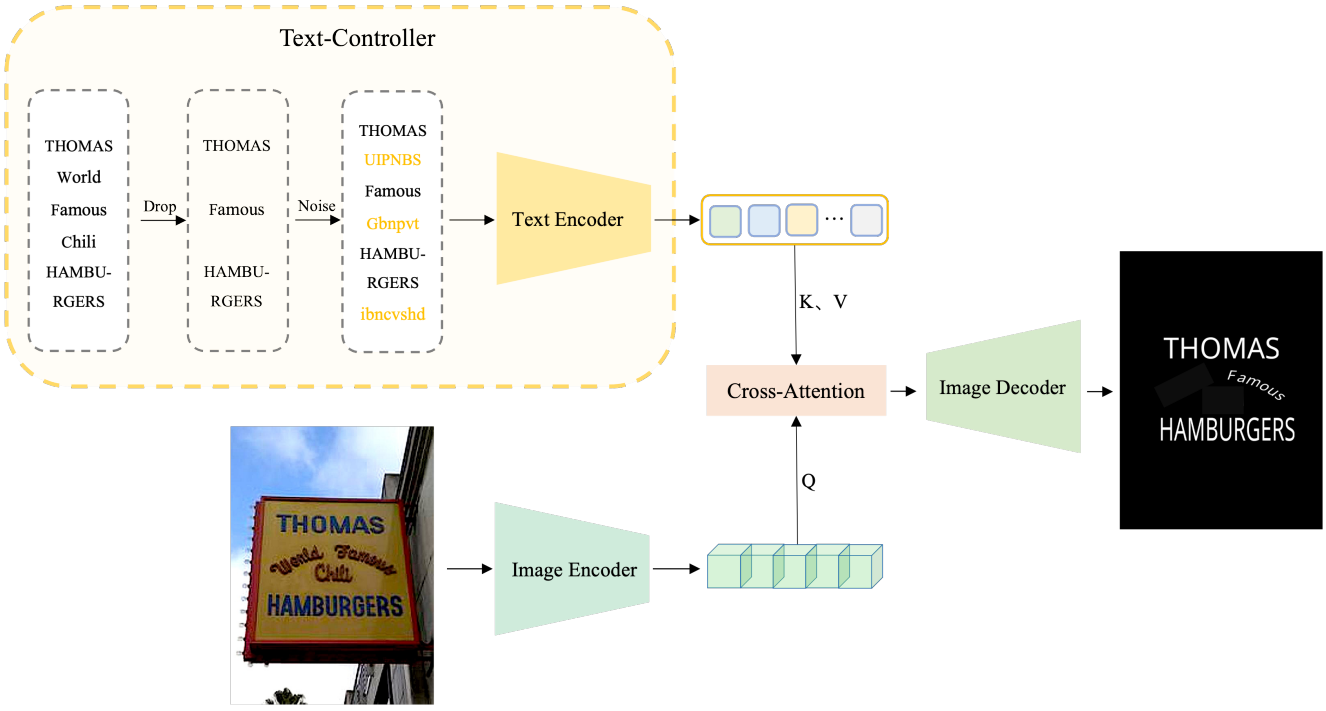

Figure 3. The overall architecture of ODM. The text is encoded by the Text-Controller to get the encoded text features, and the image is encoded by the image encoder to get the encoded image features. The text features and image features interact through cross-attention, and finally output destylization binary image.

notation formats.

Furthermore, some methods directly decode coordinates and recognition results directly by predicting sequence, utilizing the structure of transformer encoder and decoder. Both SPTS [34] and SPTS V2 [29] obtain coordinates by predicting the central point of the text instance, employing an auto-regressive approach to predict the central point and word transcription tokens. UNITS [16] can handle various types of detection formats through prompts, and they can extract text beyond the number of trained text instances. DeepSolo [54] inspired by DETR, enables the decoder to simultaneously perform text detection and recognition.

Vision-Language Pre-training. Modern pre-training methods generally involve MLM [6] or MIM [3, 10, 49], or a combination of both. The MLM task randomly masks a set of text tokens from the input and reconstructs them based on the context around the masked tokens. MIM, on the other hand, randomly masks a percentage of image patches and predicts the RGB values of raw pixels. STKM [45] learns text knowledge from datasets with image-level text annotations, the acquired text knowledge can subsequently be transferred to various text detectors. Inspired by CLIP [35], VLPT [40] adopts fine-grained cross-modality interaction to align unimodal embeddings for learning better representations of backbone via carefully designed pre-training tasks. oCLIP [52] proposes an MLM-based vision-language pre-training method, which has achieved excellent performance in text detection and text spotting tasks. StrucTexTv2 [56] implements pre-training through MIM, where it randomly masks some text word regions in the input images and feeds them into the encoder.

While these methods have shown the effectiveness of pre-training in enhancing OCR performance, they either do not utilize higher-level textual semantic information as input or do not explicitly utilize the positional information of OCR-Text, making it difficult to achieve effective alignment between text and OCR-Text. Inspired by MaskFeat [48], which achieves pre-training by reconstructing the HOG features of masked image regions, and considering that glyph has been employed in some OCR tasks [4, 33, 43, 53] with proven efficacy. We propose utilizing the Text Controller module to reconstruct the corresponding destylized glyph of the text. This approach enables visible alignment between text and OCR-Text. As shown in Fig. 4, our method can better attend to OCR-Text, highlighting its superiority in learning visual text representations for scene text image tasks.

3. Methodology

We introduce ODM, a pre-training technique that effectively aligns text and OCR-Text, leveraging the intrinsic

(a) Origin (b) No change (c) Drop-Text (d) Noise-Text

Figure 4. Illustration of the proposed Text-Controller Module: The attention heatmap (from the cross-attention layer) of the text branch under different input scenarios is depicted. (a)The original image. (b) The text input consists of three instances: "Rootin", "Ridge", and "Toymakers". (c) The "Toymakers" instance is discarded. (d) A non-existent instance "Sjehf" is added.

characteristics of OCR-Text and explicitly learning the location information. The overall pipeline of our method is depicted in Fig. 3. Our network consists of two input branches: the image encoder, which employs ResNet50 [9] to extract features from input images, and the text encoder, designed to extract textual features. By applying cross-attention between the extracted textual and visual features, we generate image features that align with textual features. These aligned features are then processed by a decoder to generate destylization binary images.

3.1. The Text-Controller Module

To address the challenge of the image encoder-decoder architecture in comprehending the significance of characters, we introduce the Text-Controller module to regulate the feature extraction of the image and align the OCR-Text features with textual features in the hidden space.

Drop-Text. General text encoders, such as CLIP [35], are designed to process text descriptions as input, with the primary objective of establishing alignment between text and image features. In the Text-Controller Module, control over the decoder is executed through prompts. Specifically, when a portion of the OCR-Text is input, the model is expected to only reconstruct that part of the binary image, treating the remaining OCR-Text as the background. During the training phase, the input OCR-Text is selected randomly, with a ratio varying from 0% to 100%. This strategy encourages the model to focus more on aligning the text with the corresponding OCR-Text.

Noise-Text. Noise, such as inaccurate labels, can potentially impact the model's training effectiveness. In contrast, we have introduced a concept termed Noise-Text, which leverages noise to augment the model's performance. This approach entails adding noise to the text encoder's input, introducing non-existent OCR-Text by altering its Token encoding value. The integration of these disruptive elements empowers the model to align the features of the text and OCR-Text more effectively, even in more intricate and challenging scenarios.

Figure 5. The upper row and lower row represent the original images and their corresponding predicted results, respectively.

In Fig. 4, we demonstrate the performance of cross-attention on images when employing different strategies in Text-Controller Module. Our approach achieves alignment between the text and corresponding OCR-Text while remaining unaffected by noisy text.

3.2. OCR-Text Destylization

To ensure that the image encoder can learn the fundamental features of OCR-Text, we have designed a simple decoder to reconstruct the destylized glyph of the text. This decoder comprises only basic FPN [24] layer upsampling and 1*1 convolution.

Fig. 5 demonstrates the predicted results on some real images using our proposed method. The training dataset only consists of SynthText [8] and does not include these real images. From the results, it can be observed that our proposed method effectively achieves OCR-Text destylization.

3.3. Loss Function

The ODM produces a binary image, conceptualizing supervised training as a pixel-level segmentation task. As a result, we optimize the model using a binary cross-entropy loss function for training:

$$\mathcal{L}_{seg} = \frac{1}{N}\sum_{i=1}^{N} -\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \quad (1)$$

where N represents the number of pixels, p_i represents the predicted result, and y_i represents the Ground Truth.

On the other hand, the training output of ODM is to generate a destylization binary image, rather than a text segmentation map that maintains proportional consistency with the original image. Consequently, relying exclusively

on pixel-level cross-entropy loss poses a challenge in effectively guiding the model to learn the destylization of characters. To optimize at the feature level, we incorporate the OCR LPIPS loss function proposed by OCR-VQGAN [36]. Specifically, we utilize a well-trained detector (Unet-VGG16[39]) to input both the Ground Truth and the model's predicted binary images. Subsequently, the $\mathcal{L}1$ Loss is computed to foster the learning of a rich latent space and the destylized character glyph images, which is defined as follows:

$$\mathcal{L}_{ocr} = \sum_{l} \frac{1}{H_l W_l} \|VGG_l(\hat{y}) - VGG_l(y)\|_1 \quad (2)$$

where H_l and W_l respectively represent the height and width of the output feature map in the l-th layer, VGG represents a well-trained detector, $\hat{y}$ represents the predicted value, and y represents the Ground Truth.

At the same time, drawing upon the batch-level contrastive loss proposed by CLIP, we expedite the mapping of text and images into the same semantic space by maximizing the similarity among positive samples and minimizing that among negative samples, which is defined as follows:

$$\begin{aligned} \mathcal{L}_{bc} = & CE(\frac{exp(I, T_+)}{\sum_{i=1}^{B} exp(I, T_i)}, y(I)) \\ & + CE(\frac{exp(T, I_+)}{\sum_{i=1}^{B} exp(T, I_i)}, y(T)) \end{aligned} \quad (3)$$

where CE represents the cross-entropy loss, I represents image features, T represents text features, I_+ represents the image feature corresponding to the current text feature, T_+ represents the text feature corresponding to the current image feature, and $y(*)$ represents the Ground Truth.

The loss function $\mathcal{L}_{\text{total}}$ is the sum of segmentation loss $\mathcal{L}_{\text{seg}}$, OCR LPIPS loss $\mathcal{L}_{\text{ocr}}$, and batch contrastive loss $\mathcal{L}_{\text{bc}}$, formulated as follows:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{seg} + \beta \mathcal{L}_{ocr} + \gamma \mathcal{L}_{bc} \quad (4)$$

where the weights α, β, and γ are empirically set to 1, 1, and 0.5, respectively.

3.4. Label Generation

One of the challenges in supervised training for ODM is the creation of data labels, particularly the acquisition of fine-grained pixel-level labels, which is both challenging and costly. To address this, we propose a method for generating pixel-level labels.

For four-point annotations, we calculate the dimensions of the quadrilateral and estimate the size and position of each character based on the number of characters. We then populate these characters using a font file, such as "NotoSans-Regular". For multi-point annotations, we employ the image synthesis approach from ABCNet [27] and utilize the Bezier curves provided by the labels to compute the curvature of the text. Similar to four-point annotations, we determine the size and position of each character from the multi-point annotations and calculate the slope using the top-left point of each character and the subsequent point. Once we obtain the slope, we adjust the orientation of the characters accordingly. Some examples of the generated labels are shown in Fig. 2.

During the process of acquiring pixel-level labels, there may be discrepancies in the spacing between the original OCR-Text and the pixel-level labels. As a result, there might be instances where individual characters in our generated labels do not align precisely with the position of OCR-Text characters in the original image. However, this discrepancy does not impact our task since our approach involves transforming the original image into a destylization binary image rather than performing pixel-by-pixel text segmentation.

4. Experiment

4.1. Datasets

In our experiments, we employ a substantial quantity of publicly accessible datasets, executing pre-training on the SynthText [8] dataset and fine-tuning on text detection and text spotting tasks utilizing datasets like ICDAR15 [15], CTW1500 [26], and TotalText [5].

SynthText is a synthetic dataset for text detection and recognition, consisting of more than 800K synthesized images. Each image is annotated with text regions and their corresponding textual content. The synthetic texts in the dataset are generated by integrating real texts into natural images. The positions, orientations, sizes, and appearances of the synthetic texts are randomly varied to simulate text in real-world scenarios.

ICDAR2015 consists of 1500 images, allocates 1000 images for training, and the remaining 500 for testing. The ICDAR2015 contains different types of text, including horizontal text and multi-oriented text.

CTW1500 is a dataset for curved text instances, consisting of 1500 images, comprising 1500 images, of which 1000 are allocated for training and the remaining 500 for testing.

TotalText is a dataset for curved text instances, consisting of 1555 images, with 1255 designated for training and the remaining 300 for testing.

4.2. Implementation Details

Training. In our proposed method, we employ ResNet50 as the image encoder, with a 6-layer transformer [44] serving as the text encoder. The reconstruction decoder is constructed from FPN and 1x1 convolutions. During the training phase, we configure the image size to 512x512, set the text length for the Text-Controller to 25, and cap the max-

Table 1. The performance of different models on ICDAR15, CTW1500, and TotalText for scene text detection is presented in the table. "PD" and "Syn" refer to the pre-training dataset and SynthText dataset, respectively. "+ODM" refers to our pre-trained model on the SynthText dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model, and Δ represents the improvement.

Method	PD	ICDAR 15			CTW1500			TotalText		
		P	R	H	P	R	H	P	R	H
FCENet[58]†	-	82.43	**88.34**	85.28	**85.7**	80.1	83.1	78.1	**84.85**	81.34
FCENet + ODM	Syn	**91.38**	82.19	**86.54**	84.92	**82.33**	**83.60**	**84.05**	80.67	**82.32**
Δ				+ 1.26			+ 0.5			+ 0.98
DBNet++[23]†	-	91.61	81.46	86.24	79.06	79.50	79.28	81.84	80.22	81.02
DBNet++ + ODM	Syn	**91.87**	**85.94**	**88.81**	**81.99**	**81.98**	**81.97**	**88.01**	**82.25**	**85.03**
Δ				+ 2.57			+ 2.69			+ 4.01

Table 2. Comparison with existing scene text pre-training techniques on PSENet. "PD" and "Syn" refer to the pre-training dataset and SynthText dataset, respectively. "+ODM" refers to our pre-trained model on the SynthText dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model, and Δ represents the improvement.

Method	PD	ICDAR 15			CTW1500			TotalText		
		P	R	H	P	R	H	P	R	H
PSENet[46]†	-	83.96	76.36	79.98	80.6	75.6	78.0	75.1	81.8	78.3
PSENet[46]	Syn	86.2	79.4	82.7	81.8	77.8	79.7	87.8	79.0	82.6
PSENet + STKM[45]	Syn	87.78	84.06	85.88	85.08	78.23	81.51	85.08	78.23	81.51
PSENet + oCLIP[52]	Syn	**88.95**	80.98	84.78	86.3	79.6	82.8	90.7	80.8	85.5
PSENet + oCLIP[52]	Web	-	-	-	**87.5**	79.9	83.5	**92.2**	82.4	**87.0**
PSENet + ODM	Syn	88.43	**85.41**	**86.90**	85.86	**85.38**	**85.62**	88.56	**83.37**	85.94
Δ				+ 6.92			+ 7.62			+ 7.64

imum number of text instances at 32. The learning rate is established at 1e-4. The network training is conducted on 8 A100 GPUs, with each card handling a batch size of 64, resulting in a total batch size of 512. The entire training process spans 100 epochs.

Fine-tuning. We executed evaluations on a variety of OCR tasks, encompassing text detection methods such as DB++ [23][1], FCENet [58][2], and PSENet [46][3], as well as text spotting techniques including ABCNet [27][4], DeepSolo [54][5], and SPTS [34][6]. In the comparative experiments, the only variable altered is the backbone weights, with all other settings maintained consistently.

Evaluation protocol. We use intersection over union (IoU) to determine whether the model correctly detects the region of text, and we calculate precision (P), recall (R), and Hmean (H) for comparison following ICDAR2015 [15].

4.3. Comparison with Detection Methods

To evaluate the effectiveness of our proposed method on text detection tasks, we conducted comparable experiments with DB++, PSENet, and FCENet. Our model underwent initial pre-training on the SynthText dataset and was subsequently fine-tuned on different datasets. The results of these experiments are shown in Tab. 1 and Tab. 2. The networks utilizing our pre-trained backbone demonstrated improvements compared to their original counterparts, with PSENet showing particularly notable improvement. This suggests that our pre-trained model weights can effectively focus on text regions within the scene, showcasing the significant potential of our pre-trained weights.

[1] https://github.com/open-mmlab/mmocr/tree/main/configs/textdet/dbnetpp
[2] https://github.com/open-mmlab/mmocr/tree/main/configs/textdet/fcenet
[3] https://github.com/open-mmlab/mmocr/tree/main/configs/textdet/psenet
[4] https://github.com/aim-uofa/AdelaiDet/blob/master/configs/BAText
[5] https://github.com/ViTAE-Transformer/DeepSolo
[6] https://github.com/shannanyinxiang/SPTS

Table 3. The performance of different models on ICDAR15 for scene text spotting is presented in the table. "PD" and "Syn" refer to the pre-training dataset and SynthText dataset, respectively. "+ODM" refers to our pre-trained model on the SynthText dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model, and Δ represents the improvement. 'D' and 'E' represent Detection and End-to-End, respectively. 'S', 'W', and 'G' donate using strong, weak, and generic lexicons, respectively.

Method	PD	D-H	E-S	E-W	E-G
ABCNet[27]†	-	85.22	78.01	73.86	68.09
ABCNet + ODM	Syn	**87.56**	**80.87**	**75.75**	**70.01**
Δ		+ 2.34	+ 2.86	+ 1.89	+ 1.92
DeepSolo[54]†	-	86.44	83.50	79.36	74.19
DeeoSolo + ODM	Syn	**88.08**	**85.83**	**80.67**	**75.2**
Δ		+ 1.64	+ 2.3	+ 1.31	+ 1.01
SPTS[34]†	-	-	77.5	70.2	65.8
SPTS + ODM	Syn	-	**78.8**	**72.1**	**67.8**
Δ			+ 1.3	+ 1.9	+ 2.0

Table 4. The performance of different models on CTW1500 for scene text spotting is presented in the table. "PD" and "Syn" refer to the pre-training dataset and SynthText dataset, respectively. "+ODM" refers to our pre-trained model on the SynthText dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model, and Δ represents the improvement. 'D' and 'E' represent Detection and End-to-End, respectively. 'N' and 'F' represent None and Full, respectively.

Method	PD	D-H	E-N	E-F
ABCNet[27]†	-	83.69	64.17	78.66
ABCNet + ODM	Syn	**85.40**	**65.06**	**79.79**
Δ		+ 1.71	+ 0.89	+ 1.13
DeepSolo[54]†	-	86.31	76.35	84.31
DeeoSolo + ODM	Syn	**86.58**	**78.07**	**85.65**
Δ		+ 0.27	+ 1.72	+ 1.34
SPTS[34]†	-	-	74.2	82.4
SPTS + ODM	Syn	-	**78.2**	**84.2**
Δ			+ 4.0	+ 1.8

4.4. Comparison with Spotting Methods

To evaluate the effectiveness of our proposed method on text spotting tasks, we conducted comparable experiments with ABCNet, DeepSolo, and SPTS, covering a wide range of task scenarios. Our model was initially pre-trained on the SynthText dataset and then fine-tuned on different datasets. The results of these experiments are presented in Tab. 3 and Tab. 4. The experimental results demonstrate that utilizing our pre-trained weights for fine-tuning significantly improves the performance of the models across various datasets. This indicates that the pre-trained weights acquired through our proposed pre-training method accurately locate the positions of OCR-Text and extract image features that capture the semantic information of the text instance.

Table 5. The performance of different models on LSVT for scene text detection is presented in the table. "PD" and "LSVT" refer to the pre-training dataset and LSVT dataset, respectively. "+ODM" refers to our pre-trained model with 400,000 pseudo-label images in the LSVT dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model, and Δ represents the improvement.

Method	PD	P	R	H
PSENet[46]†	-	64.61	72.29	68.23
PSENet + ODM	LSVT	**66.33**	**73.60**	**69.78**
Δ				+ 1.55
DBNet++[23]†	-	71.21	**79.28**	75.02
DBNet++ + ODM	LSVT	**75.81**	77.16	**76.48**
Δ				+ 1.46

4.5. Weakly Supervised Pre-training

To evaluate the effectiveness of our proposed method in aligning text with images in weakly labeled data, we assess its efficacy. We first employ PPOCRv3 [17] to generate pseudo labels (i.e., text and position in the image) on the 400,000 weakly annotated images from the LSVT [41] dataset. We then generate the corresponding pixel-level destylized annotations using our proposed label generation method. To ensure the quality of pre-training, we only select labels with inference confidence exceeding 0.9 and a text size larger than 32 pixels. We pre-train our model with these generated labels and subsequently fine-tune different scene text detectors using 30,000 fully annotated images from the LSVT dataset. The results presented in Tab. 5 demonstrate that our proposed method, when exclusively utilizing the pseudo labels for pre-training, achieves approximately a 1.5% improvement in the Hmean score after fine-tuning on both DBNet++ and PSENet models. This demonstrates that our method can perform consistently even under weakly supervised scenarios and partially addresses the issue of a large amount of unlabeled data not being able to participate in pre-training.

4.6. Comparison with Pre-training Methods

We compare our proposed method with existing scene text pre-training strategies, including STKM and oCLIP. To investigate the effectiveness of different pre-training objectives, we conducted ablative experiments with PSENet on three datasets: ICDAR15, CTW1500, and TotalText. As shown in Tab. 2, when pre-training on the same set of data, our proposed method outperforms the existing pre-training techniques. Furthermore, when fine-tuning on the

Table 6. Proportion ablation study of the proposed Text-Controller module on CTW1500. "PD" and "Syn" refer to the pre-training dataset and SynthText dataset, respectively. "TP" refers to the selected text proportion. "+ODM" refers to our pre-trained model on the SynthText dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model.

Method	PD	TP	P	R	H
PSENet[46]	-	-	80.6	75.6	78.0
PSENet[46]	Syn	-	81.8	77.8	79.7
PSENet + ODM	Syn	0%	82.41	84.19	83.29
PSENet + ODM	Syn	30%	84.83	82.18	83.48
PSENet + ODM	Syn	50%	83.76	**84.78**	**84.27**
PSENet + ODM	Syn	70%	**85.32**	82.37	83.82

Table 7. Ablation study of our proposed components on TotalText. We fine-tune PSENet by using the pre-trained models with different modules. "TE", "DT", "NT", and "OL" refer to Text Encoder, Drop-Text, Noise-Text, and OCR Loss, respectively.

Method	TE	DT	NT	OL	P	R	H
PSENet[46]					75.10	81.80	78.30
PSENet + ODM					85.08	78.55	81.68
PSENet + ODM	✓				86.66	82.75	84.66
PSENet + ODM	✓	✓			88.06	82.97	85.44
PSENet + ODM	✓		✓		88.10	82.57	85.24
PSENet + ODM	✓	✓	✓		88.17	83.21	85.62
PSENet + ODM	✓	✓	✓	✓	**88.56**	**83.37**	**85.94**

CTW1500 dataset, our proposed method, which was pre-trained on SynthText alone, even surpasses the performance of oCLIP, which was pre-trained on 40 million web images.

4.7. Ablation Experiments

Proportion Ablation We conducted experiments to assess the influence of selected proportions in our Drop-Text and Noise-Text strategies. Four groups of experiments were performed with selected proportions of 0%, 30%, 50%, and 70% respectively. The selected proportions for these strategies were the only variables adjusted, while all other configurations remained constant. We pre-trained the models on the SynthText dataset with varying proportions and transferred the pre-trained weights to fine-tune PSENet on the CTW1500 dataset. The results are presented in Tab. 6. The performance of the model was effectively enhanced by facilitating text-image alignment through the Drop-Text and Noise-Text strategies. The impact of varying proportions on performance was substantial. A small proportion of text instances resulted in minimal performance improvement, possibly due to the insufficient changes in a small number of words to achieve effective text-image alignment. This minimal improvement can be attributed to the errors introduced during training due to the lack of significant changes in the text instances. On the other hand, using a large proportion of text instances also had a limited effect on performance. This can be attributed to the fact that when a substantial number of words are either eliminated or added as noise, the model captures fewer valid features during training, leading to difficulties in convergence and potentially biased learning. Hence, it is important to select an appropriate proportion that ensures the model captures an adequate number of valid features. This helps the model align text with OCR-Text in more complex scenarios and enhances the robustness of the model.

Module Ablation: In our research, we investigated the contributions of several proposed modules. We trained the models with different combinations of these modules on the SynthText dataset and subsequently fine-tuned the pre-trained weights using PSENet on the TotalText dataset, as shown in Tab. 7. The empirical results indicated that for the task of OCR-Text Desytlization, the text feature furnished by the Text-Controller module aids the model in better understanding and locating OCR-Text. Additionally, our proposed Drop-Text and Noise-Text strategies effectively bolster the model's performance by intensifying the alignment between text and OCR-Text.

5. Conclusion

This paper introduces ODM, a novel pre-trained method that aims to transform diverse styles of text found in images into a uniform style based on the text prompt. By leveraging the pre-trained model generated through ODM, it can seamlessly integrate into existing detection and spotting networks, resulting in significant performance improvements. To address the challenge of annotation costs in OCR tasks, we propose a new labeling generation method designed specifically for ODM. Additionally, we introduce the Text-Controller module, which helps regulate the model output and improves its understanding of OCR-Text. By combining these approaches, we enable a larger amount of unlabeled data to be used in the pre-training process, effectively reducing annotation costs. Our extensive ablation and comparative experiments demonstrate the effectiveness and robustness of our model. The results highlight the potential of ODM in OCR pre-training and its valuable contributions to the advancement of scene text detection and spotting tasks. Looking ahead, we plan to explore the potential of this method in other domains, such as document analysis, handwriting recognition, and other complex scene text scenarios.

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[2] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD in-*

ternational conference on knowledge discovery & data mining, pages 71–79, 2018.

[3] Jun Chen, Ming Hu, Boyang Li, and Mohamed Elhoseiny. Efficient self-supervised vision pretraining with local masked reconstruction. *arXiv preprint arXiv:2206.00790*, 2022.

[4] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36, 2024.

[5] Chee-Kheng Ch'ng, Chee Seng Chan, and Cheng-Lin Liu. Total-text: toward orientation robustness in scene text detection. *International Journal on Document Analysis and Recognition (IJDAR)*, 23(1):31–52, 2020.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Textdragon: An end-to-end framework for arbitrary shaped text spotting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9076–9085, 2019.

[8] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Mae: Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[11] Minghang He, Minghui Liao, Zhibo Yang, Humen Zhong, Jun Tang, Wenqing Cheng, Cong Yao, Yongpan Wang, and Xiang Bai. Most: A multi-oriented scene text detector with localization refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8813–8822, 2021.

[12] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5020–5029, 2018.

[13] Mingxin Huang, Yuliang Liu, Zhenghao Peng, Chongyu Liu, Dahua Lin, Shenggao Zhu, Nicholas Yuan, Kai Ding, and Lianwen Jin. Swintextspotter: Scene text spotting via better synergy between text detection and text recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4593–4603, 2022.

[14] Mingxin Huang, Jiaxin Zhang, Dezhi Peng, Hao Lu, Can Huang, Yuliang Liu, Xiang Bai, and Lianwen Jin. Estextspotter: Towards better scene text spotting with explicit synergy in transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19495–19505, 2023.

[15] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.

[16] Taeho Kil, Seonghyeon Kim, Sukmin Seo, Yoonsik Kim, and Daehee Kim. Towards unified scene text spotting based on sequence generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15223–15232, 2023.

[17] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, et al. Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system. *arXiv preprint arXiv:2206.03001*, 2022.

[18] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[19] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[20] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 27(8):3676–3690, 2018.

[21] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. 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*, pages 706–722. Springer, 2020.

[22] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11474–11481, 2020.

[23] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):919–931, 2022.

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[25] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5676–5685, 2018.

[26] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 90: 337–345, 2019.

[27] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9809–9818, 2020.

[28] Yuliang Liu, Chunhua Shen, Lianwen Jin, Tong He, Peng Chen, Chongyu Liu, and Hao Chen. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8048–8064, 2021.

[29] Yuliang Liu, Jiaxin Zhang, Dezhi Peng, Mingxin Huang, Xinyu Wang, Jingqun Tang, Can Huang, Dahua Lin, Chunhua Shen, Xiang Bai, et al. Spts v2: single-point scene text spotting. *arXiv preprint arXiv:2301.01635*, 2023.

[30] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 67–83, 2018.

[31] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2): 532–548, 2021.

[32] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE transactions on multimedia*, 20(11):3111–3122, 2018.

[33] Jian Ma, Mingjun Zhao, Chen Chen, Ruichen Wang, Di Niu, Haonan Lu, and Xiaodong Lin. Glyphdraw: Learning to draw chinese characters in image synthesis models coherently. *arXiv preprint arXiv:2303.17870*, 2023.

[34] Dezhi Peng, Xinyu Wang, Yuliang Liu, Jiaxin Zhang, Mingxin Huang, Songxuan Lai, Jing Li, Shenggao Zhu, Dahua Lin, Chunhua Shen, et al. Spts: single-point text spotting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4272–4281, 2022.

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[36] Juan A Rodriguez, David Vazquez, Issam Laradji, Marco Pedersoli, and Pau Rodriguez. Ocr-vqgan: Taming text-within-image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3689–3698, 2023.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[38] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2550–2558, 2017.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[40] Sibo Song, Jianqiang Wan, Zhibo Yang, Jun Tang, Wenqing Cheng, Xiang Bai, and Cong Yao. Vision-language pre-training for boosting scene text detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15681–15691, 2022.

[41] Yipeng Sun, Zihan Ni, Chee-Kheng Chng, Yuliang Liu, Canjie Luo, Chun Chet Ng, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, et al. Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1557–1562. IEEE, 2019.

[42] Jingqun Tang, Wenqing Zhang, Hongye Liu, MingKun Yang, Bo Jiang, Guanglong Hu, and Xiang Bai. 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*, pages 4563–4572, 2022.

[43] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. *arXiv preprint arXiv:2311.03054*, 2023.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[45] Qi Wan, Haoqin Ji, and Linlin Shen. Self-attention based text knowledge mining for text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5983–5992, 2021.

[46] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9336–9345, 2019.

[47] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8440–8449, 2019.

[48] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022.

[49] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022.

[50] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Transactions on Image Processing*, 28(11):5566–5579, 2019.

[51] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training

for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020.

[52] Chuhui Xue, Wenqing Zhang, Yu Hao, Shijian Lu, Philip HS Torr, and Song Bai. Language matters: A weakly supervised vision-language pre-training approach for scene text detection and spotting. In *European Conference on Computer Vision*, pages 284–302. Springer, 2022.

[53] Mingkun Yang, Biao Yang, Minghui Liao, Yingying Zhu, and Xiang Bai. Class-aware mask-guided feature refinement for scene text recognition. *Pattern Recognition*, 149:110244, 2024.

[54] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Tongliang Liu, Bo Du, and Dacheng Tao. Deepsolo: Let transformer decoder with explicit points solo for text spotting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19348–19357, 2023.

[55] Wenwen Yu, Yuliang Liu, Wei Hua, Deqiang Jiang, Bo Ren, and Xiang Bai. Turning a clip model into a scene text detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6978–6988, 2023.

[56] Yuechen Yu, Yulin Li, Chengquan Zhang, Xiaoqiang Zhang, Zengyuan Guo, Xiameng Qin, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. Structextv2: Masked visual-textual prediction for document image pre-training. *arXiv preprint arXiv:2303.00289*, 2023.

[57] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.

[58] Yiqin Zhu, Jianyong Chen, Lingyu Liang, Zhanghui Kuang, Lianwen Jin, and Wayne Zhang. Fourier contour embedding for arbitrary-shaped text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3123–3131, 2021.