Turning a CLIP Model Into a Scene Text Spotter

Wenwen Yu, Yuliang Liu, *Member, IEEE*, Xingkui Zhu, Haoyu Cao, Xing Sun, and Xiang Bai, *Senior Member, IEEE*

***Abstract*—We exploit the potential of the large-scale Contrastive Language-Image Pretraining (CLIP) model to enhance scene text detection and spotting tasks, transforming it into a robust backbone, FastTCM-CR50. This backbone utilizes visual prompt learning and cross-attention in CLIP to extract image and text-based prior knowledge. Using predefined and learnable prompts, FastTCM-CR50 introduces an instance-language matching process to enhance the synergy between image and text embeddings, thereby refining text regions. Our Bimodal Similarity Matching (BSM) module facilitates dynamic language prompt generation, enabling offline computations and improving performance. FastTCM-CR50 offers several advantages: 1) It can enhance existing text detectors and spotters, improving performance by an average of 1.6% and 1.5%, respectively. 2) It outperforms the previous TCM-CR50 backbone, yielding an average improvement of 0.2% and 0.55% in text detection and spotting tasks, along with a 47.1% increase in inference speed. 3) It showcases robust few-shot training capabilities. Utilizing only 10% of the supervised data, FastTCM-CR50 improves performance by an average of 26.5% and 4.7% for text detection and spotting tasks, respectively. 4) It consistently enhances performance on out-of-distribution text detection and spotting datasets, particularly the NightTime-ArT subset from ICDAR2019-ArT and the DOTA dataset for oriented object detection.**

***Index Terms*—CLIP, few-shot, generalization, rotated object, scene text detection, scene text spotting.**

I. Introduction

SCENE text spotting, aiming at the localization and recognition of text instances within natural images, has remained at the forefront due to its diverse practical applications, which include online education, office automation, automatic driving, and instant translation. The evolution of fully-supervised deep learning technologies has spearheaded substantial advancements within scene text spotting. Yet, these supervised methodologies, are heavily reliant on detailed and extensive annotations, indicating a potential limitation when facing scenarios with divergent data distributions. How to improve the performance of text spotting techniques under circumstances of sparse annotated data or when shifting between different domains - commonly referred to as few-shot training and generalization ability - is increasingly gaining attention.

In the past decade, utilizing the backbones such as VGG16 and ResNet-50 from ImageNet and MSCOCO to acquire better initialization and generalization ability for scene text detection and spotting are commonly adopted as a basic setting. Recently, developments in leveraging pretrained vision and language knowledge, particularly through the large-scale Contrastive Language-Image Pretraining (CLIP) model [1], have shown promising results in a wide range of downstream tasks. These include but are not limited to image classification [2], object detection [3], and semantic segmentation [4], [5]. In the realm of text spotting, where scene text often provides rich visual and character information, the potential of the CLIP model is particularly evident. How to excavate cross-modal information from visual, semantic, and text knowledge to enhance the performance of text detection and spotting models has gained more and more attention. Song et al. [6], for instance, has proposed a fine-grained cross-modality interaction approach, inspired by CLIP, to align unimodal embeddings and improve the learning of representations through pretext task pretraining for scene text detection. Wan et al. [7] have brought forth an approach that involves a self-attention based text knowledge mining technique to boost the backbone via image-level text recognition pretext task pretraining. Meanwhile, Xue et al. [8] have introduced a weakly supervised pretext task pretraining method aiming to jointly learn and align visual and partial textual information. The goal is to cultivate effective visual text representations applicable to scene text detection and spotting.

Contrary to existing approaches illustrated in Fig. 2, our aim is to transform the CLIP model directly into a foundation for text detection and spotting, eliminating the need for pretext task pretraining process. However, this is not a straightforward task, as we empirically observe that only solely employing the CLIP model leads to minimal enhancements, and even worse results in aerial object detection, as shown in Section IV-I. The primary challenge lies in identifying an effective method to leverage visual and semantic prior information specific to each image.

To this end, we introduce a new backbone specifically designed for scene text detection and spotting tasks, termed as *FastTCM-CR50*. This model can be conveniently incorporated into existing scene text detection and spotting frameworks to enhance their performance. Central to our approach is a

Manuscript received 30 July 2023; revised 13 February 2024; accepted 10 March 2024. Date of publication 20 March 2024; date of current version 6 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62225603 and Grant 62206103, and in part by National Key Research and Development Program under Grant 2022YFC2305102. Recommended for acceptance by V. Morariu. *(Corresponding author: Yuliang Liu.)*

Wenwen Yu, Yuliang Liu, Xingkui Zhu, and Xiang Bai are with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: wenwenyu@hust.edu.cn; ylliu@hust.edu.cn; adlith@hust.edu.cn; xbai@hust.edu.cn).

Haoyu Cao and Xing Sun are with Tencent, YouTu Lab, Hefei 230000, China (e-mail: rechycao@tencent.com; winfred.sun@gmail.com).

The code is available at https://github.com/wenwenyu/TCM.

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3379828, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3379828

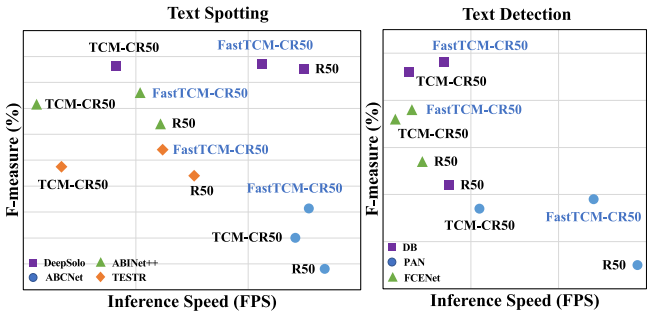


Fig. 1. Comparison of F-measure and FPS among different backbones on text spotting and text detection methods. FastTCM-CR50 achieves higher performance while significantly improving the inference speed compared to TCM-CR50. The text spotting F-measure is calculated across IC15, Total-Text, and CTW1500. The text detection F-measure is calculated across IC15, TD, and CTW1500, both averaged across the datasets.

cross-modal interaction mechanism established through visual prompt learning. The mechanism, realized via cross-attention, recovers the locality feature from the CLIP image encoder, thereby capturing fine-grained information for the subsequent matching of text instances with the language, which is particularly useful in responding to coarse text regions. Further, to exploit the prior knowledge from the text encoder for different input images, we utilize an improved language prompt unit built on the learnable query and bimodal similarity matching to obtain global image information. In addition, we have also devised an instance-language matching method to align the image and text embeddings, which aids the image encoder to refine text regions based on cross-modal visual-language priors. The FastTCM-CR50 model can then be directly fine-tuned for the text detection and spotting tasks without requiring a pretext task pretraining, as detailed in Fig. 2(c). Compared to our conference version TCM-CR50 [9], FastTCM-CR50 introduces Bimodal Similarity Matching (BSM) module as well as the learnable parameters as an implicit image condition that enables and further enhances the CLIP text encoder to perform offline calculations during inference, thereby achieving better results and reducing the inference time, as shown in Fig. 1.

We summarize the advantages of our method as follows:

- Our proposed FastTCM-CR50 backbone inherently enhances current scene text detectors and spotters, resulting in average performance improvements with numerous baseline methods by 1.6% and 1.5% for scene text detection and spotting tasks, respectively.
- Besides, FastTCM-CR50 outperforms the previous text detection and spotting backbone TCM-CR50, delivering an average performance boost of 0.2% and 0.55% in text detection and spotting tasks, respectively, along with a notable 47.1% increase in inference speed.
- Demonstrating robust few-shot training capabilities, our new backbone, when trained with only 10% of the supervised data, exhibits an impressive average performance surge of 26.5% and 4.7% for text detection and spotting tasks, respectively.
- In terms of generalization ability, our approach notably surpasses baseline methods by an average of 12.4% and 14.8% for domain adaptation tasks for text detection and spotting, respectively. Particularly noteworthy are the significant improvements achieved on the NightTime-ArT subset from ICDAR2019-ArT and the rotated object detection dataset, DOTA-v1.0, illustrating its robust generalization capabilities across diverse task domains.

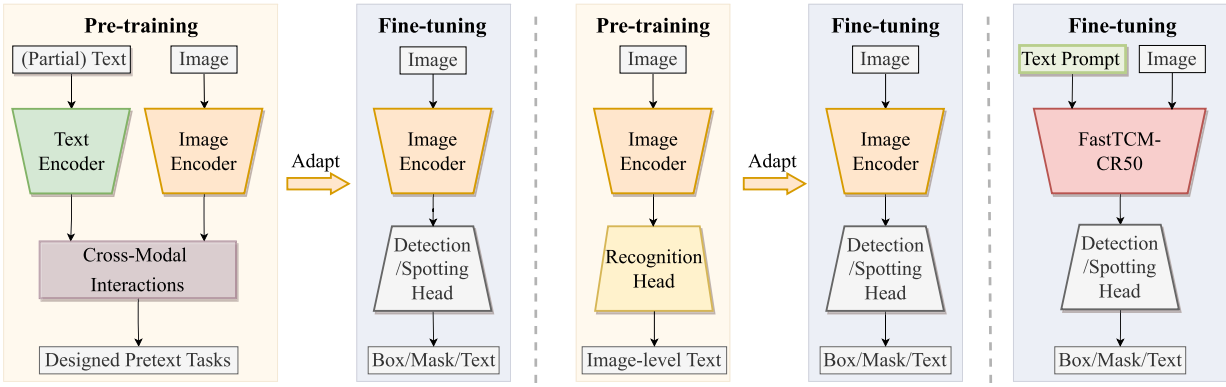
II. Related Works

A. Scene Text Detection

Scene text detection is a technique that exclusively utilizes bounding box annotations. This method can generally be categorized into two primary types: segmentation-based and regression-based techniques.

Segmentation-based Methods: Segmentation-based techniques typically perform operations at the pixel, segment, or contour level, subsequently grouping these into text instances. Notable methods include the Segment Linking (SegLink) by Shi et al. [10], using a fully-convolutional neural network for detecting segments and links; the TextSnake by Long et al. [11], an adaptable approach for detecting text of varying shapes; and the Progressive Scale Expansion Network (PSENet) by Li et al. [12], generating diverse kernel scales for each text instance. Efficient and accurate systems like the Pixel Aggregation Network (PAN) developed by Wang et al. [13] have emerged, combining a low computational-cost segmentation head with a learnable post-processing system. Additionally, unique methods such as the Differentiable Binarization (DB) module introduced by Liao et al. [14] incorporate the binarization step directly into the segmentation network. Meanwhile, the transformer-based architecture proposed by Tang et al. [15] performs detection based on select representative features to decrease computational cost and reduce background interference. These methods underscore the wide range and adaptability of segmentation-based techniques in text detection. Further, Long et al. [16] introduced an end-to-end model capable of performing unified scene text detection and visual layout analysis simultaneously.

Regression-based Methods: Regression-based methods view text as a single object and directly regress the bounding boxes of the text instances. Zhang et al. [17] propose a multi-oriented text detection method utilizing Fully Convolutional Networks, which uses both local and global cues to locate text lines. Liu et al. [18] develop the Deep Matching Prior Network (DMPNet), using quadrilateral sliding windows and a sequential protocol for regression to predict text with a compact quadrangle. He et al. [19], [20] introduced models for text detection that utilize a regional attention mechanism and deep direct regression to predict the text bounding box. Liao et al. [21] create a unified deep neural network for natural image text detection, and Zhou et al. [22] designed the EAST model that predicts words or text lines of any orientation and quadrilateral shape in full images. Innovative methods like LOMO by Zhang et al. [23], and the adaptive text region representation by Wang et al. [24] have also been developed. Zhu et al.'s FCENet [25], Liu et al.'s MOST [26], and Zhang et al.'s adaptive boundary proposal network [27] further contribute to the field by introducing novel concepts and methodologies. Dai et al. [28] use a progressive contour



(a) Pretraining with cross-modal interactions (b) Pretraining with pretext tasks (c) Ours

Fig. 2. We compare different paradigms of utilizing text knowledge for scene text detection and spotting. Our approach directly delivers an enhanced CLIP backbone, eliminating the need for a pretraining process that relies on specifically designed pretext tasks. CR50 represents for CLIP-ResNet50 model.

regression strategy, and Ye et al.'s DPText-DETR [29] employs explicit point coordinates and an enhanced self-attention module. Zhang et al. [30] present a unified coarse-to-fine framework for text detection using an iterative boundary transformer.

B. Scene Text Spotting

Scene text spotting typically employs a unified end-to-end trainable network, blending text detection and text recognition into a cross-modal assisted paradigm. This integrated approach streamlines text detection and recognition into a singular network. It enables simultaneous localization and identification of text within images, capitalizing on the synergistic relationship between text detection and recognition to augment overall performance. Scene text spotting can be bifurcated into two principal categories: regular end-to-end scene text spotting and arbitrarily-shaped end-to-end scene text spotting. Regular end-to-end scene text spotting concentrates on discerning and deciphering text within standard-shaped regions, whereas arbitrarily-shaped end-to-end scene text spotting broadens its scope to manage text in irregular or curved formations.

Regular End-to-end Scene Text Spotting: Li et al. [31] propose one of the earliest end-to-end trainable scene text spotting methods. Their approach effectively merged detection and recognition features using RoI Pooling [32] in a two-stage framework. Originally designed for horizontal and focused text, their method showed significant performance improvements in an enhanced version [33]. Busta et al. [34] made contributions to the field with their end-to-end deep text spotter. In further advancements, He et al. [35] and Liu et al. [36] incorporated anchor-free mechanisms to enhance training and inference speed. They employed different sampling strategies, such as Text-Align-Sampling and RoI-Rotate, respectively, to extract features from quadrilateral detection results.

Arbitrarily-shaped End-to-end Scene Text Spotting: Liao et al. [37] introduced Mask TextSpotter which uses Mask R-CNN with character-level supervision to detect and recognize arbitrarily-shaped text. Mask TextSpotterv2 [38] reduces reliance on character-level annotations. Qin et al. [39] employ RoI Masking for attention on arbitrarily-shaped text regions. Feng et al. [40] utilize RoISlide for handling long text, whereas Wang et al. [41] focus on boundary points detection, text rectification, and recognition. CharNet by Xing et al. [42] also caters to arbitrarily-shaped text spotting. Liao et al.'s Segmentation Proposal Network (SPN) [43] and Liu et al.'s ABCNet [44] are other noteworthy contributions. ABINet++ by Fang et al. [45] innovatively uses a vision model and a language model with an iterative correction mechanism. Huang et al.'s SwinTextSpotter [46] uses a transformer encoder for detection and recognition. Approaches based on DETR [47] and variants [48] for RoI-free scene text spotting have also shown promising results. TESTR [49] uses an encoder and dual decoders, while TTS [50] uses a transformer-based approach. SPTS [51] employs a single point for each instance and uses a Transformer to predict sequences. DeepSolo [52] allows a single decoder to perform text detection and recognition.

C. Cross-Modal Pretraining Methods

Cross-modal assisted methods leverage a rich blend of cross-modal information including visual, semantic, and text data to amplify the performance for scene text detection and spotting tasks. Wan et al. [7], for instance, implemented image-level text recognition pretext task pretraining to fortify the backbone using their proposed self-attention-based text knowledge mining mechanism. Taking inspiration from CLIP, Song et al. [6] formulated three pretext task pretraining for fine-grained cross-modality interaction, designed to align unimodal embeddings and learn enhanced representations of the backbone. Xue et al. [8] proposed a weakly supervised pretext task pretraining method, which simultaneously learns and aligns visual and partial text instance information, with the aim of producing effective visual text representations.

D. Comparison to the Conference Version

This paper is a substantial extension of our prior publication [9]. Building upon this foundation, our current study incorporates three major improvements that contribute to the

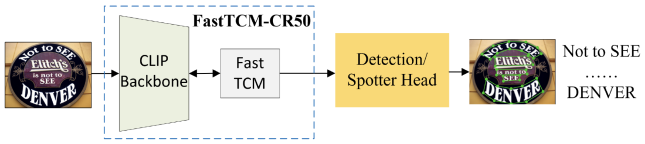


Fig. 3. Overall framework of our approach.

advancement of the field of scene text detection and scene text spotting.

1) We introduced FastTCM-CR50, an innovative text spotting backbone that overcomes the limitations of our conference version which is solely tested on scene text detection. It incorporates a meta query and Bimodal Similarity Matching (BSM), eliminating the need for text encoder in the inference process, leading to a remarkable speedup. Moreover, it dynamically augments text embeddings with visual modalities, enhancing the overall performance. Specifically, it brings about substantial improvements in inference speed (by 47.1%) while enhancing performance.
2) Extensive experiments were conducted to evaluate the performance of TCM and FastTCM in different settings. We explored their utility in boosting the efficacy of existing text detectors and spotters, their competence in few-shot learning, and their domain adaptation capabilities. Our thorough ablation studies offered insights into the contributions of our method in harnessing pretrained CLIP knowledge to elevate the performance of text detectors and spotters.
3) Our method exhibited impressive adaptability across diverse tasks. The proposed FastTCM-CR50 showed their efficacy in scene text spotting and complex tasks like oriented, dense, and small object detection in aerial imagery.

III. Methodology

An overview of our approach is shown in Fig. 3. In essence, we repurpose the CLIP model to serve as the backbone, utilizing the FastTCM as a bridge between the CLIP backbone and the detection/spotter heads.

A. Prerequisite: CLIP Model

The CLIP model [1] has demonstrated substantial potential in the realm of learning transferable knowledge and open-set visual concepts, given its capacity to analyze 400 million unannotated image-text pairs during its pretraining phase. Prior research [53] reveals that CLIP's individual neurons are adept at capturing concepts in literal, symbolic, and conceptual manners, which serves as an innately text-friendly model, capable of effectively mapping the space between image and text [54]. During its training phase, CLIP learns a joint embedding space for two modalities through a contrastive loss. Given a batch of image-text pairs, the model maximizes the cosine similarity with matching text and minimizes the similarity with all other unmatched text for each image. The same process applies to each piece of text, which has allowed CLIP to be utilized for zero-shot image recognition [2]. However, leveraging the valuable insights generated by such a model presents two prerequisites. First, an effective method is required to access the prior knowledge stored within the CLIP model. Second, while the original model is designed to measure the similarity between a complete image and a single word or sentence, scene text detection and spotting usually involve numerous text instances per image, all of which need to be equivalently recalled.

B. FastTCM

FastTCM, designed to enhance the CLIP model, serves as a robust foundation for boosting existing scene text detectors and spotters. It achieves this by extracting both image and text embeddings from CLIP's image and text encoders, respectively. The first step in the process is designing a cross-modal interaction mechanism. We do this via visual prompt learning which restores the locality feature from CLIP's image encoder. The enhanced locality feature allows for capturing fine-grained data to effectively respond to a more general text region, setting the stage for subsequent matches between text instances and language. Next, to better channel pre-trained knowledge, we build a language prompt unit. This unit produces a contextual cue for each image. For the efficient extraction of interactions between the image and text encoder, all while enabling faster inferences, we use a method called Bimodal Similarity matching. This method allows for the offline computation of inferences using the CLIP text encoder, along with the dynamic generation of language prompts that are based on the conditions of the image. Finally, an instance-language matching technique is employed to align the image and text embeddings. This encourages the image encoder to meticulously refine text regions from the cross-modal visual-language priors.

1) Image Encoder: We use the pretrained ResNet50 [55] of CLIP as the image encoder, which produces an embedding vector for every input pixel. Given the input image $\boldsymbol{I}' \in \mathbb{R}^{H\times W\times 3}$, image encoder outputs image embedding $\boldsymbol{I} \in \mathbb{R}^{\tilde{H}\times\tilde{W}\times C}$, where $\tilde{H} = \frac{H}{s}, \tilde{W} = \frac{W}{s}$, and C is the image embedding dimension (C is set to 1024) and s is the downsampling ratio (s is empirically set to 32), which can be expressed as:

$$\boldsymbol{I} = \text{ImageEncoder}(\boldsymbol{I}')\,. \tag{1}$$

2) Text Encoder: The text encoder takes input a number of of K classes prompt and embeds it into a continuous vector space $\mathbb{R}^C$, producing text embeddings $\boldsymbol{T} = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_K\} \in \mathbb{R}^{K\times C}$ as outputs of the text encoder, where $\boldsymbol{t}_i \in \mathbb{R}^C$. Specifically, we leverage the frozen pretrained text encoder of CLIP throughout as the text encoder can provide language knowledge prior to text detection and spotting. K is set to 1 because there is only one text class in text detection task. Different from the original model that uses templates like "a photo of a [CLS].", we predefine discrete language prompt as "*Text*". Then, a part of the text encoder input $\boldsymbol{t}'_{in}$ is defined as follows:

$$\boldsymbol{t}'_{in} = \text{WordEmbedding}(\text{Text}) \in \mathbb{R}^{\text{D}}\,, \tag{2}$$

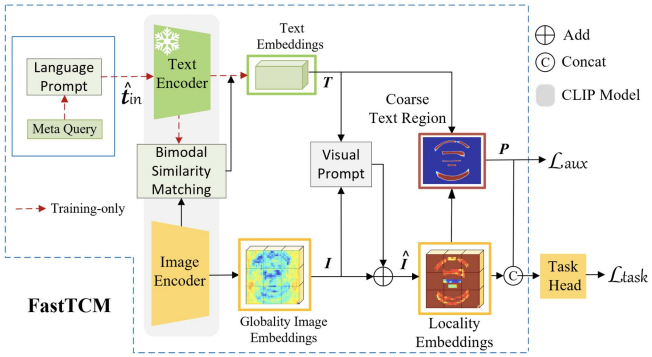


Fig. 4. Details of the FastTCM. The image encoder and text encoder are directly from the CLIP model. The red dashed arrows represent training-only operators, with the corresponding upstream calculation procedure performed offline during the inference stage.

where $\mathrm{WordEmbedding}(\cdot)$ denotes word embedding for predefined prompt "Text" class. D is the word embedding dimension and is set to 512.

Inspired by CoOp [2], [56], we also add learnable prompt $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n\}$ to learn robust transferability of text embedding for facilitating zero-shot transfer of CLIP model, where n is the number of learnable prompt, which is set to 4 by default, and $\boldsymbol{c}_i \in \mathcal{R}^D$. Thus, the input $\boldsymbol{t}_{in}$ of the text encoder is as follows:

$$\boldsymbol{t}_{in} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{t}'_{in}] \in \mathbb{R}^{(n+1)\times D}. \tag{3}$$

The text encoder takes $\boldsymbol{t}_{in}$ as input and generates text embedding $\boldsymbol{T} = \{\boldsymbol{t}_1\} \in \mathbb{R}^C$, and $\boldsymbol{T}$ is donated by $\boldsymbol{t}_{\text{out}} \in \mathcal{R}^C$ for simplification:

$$\boldsymbol{t}_{\text{out}} = \mathrm{TextEncoder}(\boldsymbol{t}_{in}) \in \mathbb{R}^C. \tag{4}$$

3) Language Prompt Unit: Although the predefined prompt and learnable prompt are effective for steering the CLIP model, it may suffer from limited few-shot or generalization ability to open-ended scenarios where the testing text instance is out-of-distribution from the training images. To this end, we present a language prompt module to generate a feature vector, termed as conditional cue ($\boldsymbol{cc}$), as depicted in Fig. 5. For each image, the $\boldsymbol{cc}$ is then combined with the input of the text encoder $\boldsymbol{t}_{in}$, formulated as follows:

$$\hat{\boldsymbol{t}}_{in} = \boldsymbol{cc} + \boldsymbol{t}_{in} \in \mathbb{R}^{(n+1)\times D}, \tag{5}$$

where $\hat{\boldsymbol{t}}_{in}$ is the new prompt input of the text encoder conditioned on the input image, and we replace $\boldsymbol{t}_{in}$ with $\hat{\boldsymbol{t}}_{in}$ in (4).

As depicted in Fig. 4, we introduce meta query to generate an implicit conditional cue ($\boldsymbol{cc}$) followed by a two-layer feed-forward network, enabling the decoupling of the text encoder from the inference process. In addition, we design a bimodal similarity matching (BSM) module to act as a gate, which controls the amount of visual modal information that should compensate for text modal embeddings. This dynamic enrichment of text embeddings with visual information is helpful to the overall performance of the model.

Meta Query: Specifically, FastTCM first incorporates a meta query, denoted as $\boldsymbol{MQ}$, which is initialized with learnable parameters representing the shape of $\mathbb{R}^{\mathbb{C}}$. The meta query serves as an implicit image condition to guide the generation of subsequent language prompt, steering the pretrained knowledge from the text encoder. This operation is motivated by DETR [47], which utilizes a Transformer Encoder, and Decoder that looks for a specific number of object queries (potential object detections). This substitution allows us to generate an implicit conditional cue $\boldsymbol{cc}$, and is formulated as follows:

$$\boldsymbol{cc} = \mathrm{LN}(\sigma(\mathrm{LN}(\boldsymbol{MQ})\boldsymbol{W}_1 + \boldsymbol{b}_1))\boldsymbol{W}_2 + \boldsymbol{b}_2 \in \mathbb{R}^D, \tag{6}$$

where $\boldsymbol{cc}$ represents the generated implicit conditional cue, which is utilized in subsequent steps. $\boldsymbol{W}_1 \in \mathbb{R}^{C\times C}$, $\boldsymbol{W}_2 \in \mathbb{R}^{C\times D}$, $\boldsymbol{b}_1 \in \mathbb{R}^C$, $\boldsymbol{b}_2 \in \mathbb{R}^D$, and we broadcast $\boldsymbol{cc}$ with $\boldsymbol{t}_{in}$ to get $\hat{\boldsymbol{t}}_{in}$ in (5). It is important to note that once training is completed, the meta query remains unchanged. This allows the CLIP text encoder to perform offline participant calculation during inference, resulting in reduced inference time and making FastTCM more suitable for practical real-world applications.

Bimodal Similarity Matching: Given the output of the text encoder $\boldsymbol{t}_{\text{out}}$ and the global image-level feature $\bar{\boldsymbol{I}}$, we first calculate the cosine similarity between text embeddings and globality image, as defined by the following equation:

$$\mathrm{sim} = \frac{\bar{\boldsymbol{I}} \cdot \boldsymbol{t}_{\text{out}}}{|\bar{\boldsymbol{I}}||\boldsymbol{t}_{\text{out}}|}, \tag{7}$$

where sim serves as the relevance threshold for an output gate that controls the amount of visual modal information used to compensate for text modal embeddings. Next, using the relevance threshold sim, we apply a weighted sum between $\hat{\boldsymbol{t}}_{\text{out}}$ and $\bar{\boldsymbol{I}}$ as follows:

$$\hat{\boldsymbol{t}}_{\text{out}} = \mathrm{sim} \cdot \bar{\boldsymbol{I}} + \boldsymbol{t}_{\text{out}}, \tag{8}$$

where $\hat{\boldsymbol{t}}_{\text{out}}$ is the new output of the text encoder, which is dynamically post-conditioned on the implicit image features. We use $\hat{\boldsymbol{t}}_{\text{out}}$ to replace $\boldsymbol{t}_{\text{out}}$ in subsequent processes, including visual prompt generator (9) and instance-language matching (11).

4) Visual Prompt Generator: We design a visual prompt generator to adaptively propagate fine-grained semantic information from textual features to visual features, as presented in Fig. 5. Formally, we use the cross-attention mechanism in Transformer [57] to model the interactions between image embedding ($\boldsymbol{Q}$) and text embedding ($\boldsymbol{K}$, $\boldsymbol{V}$). The visual prompt $\tilde{\boldsymbol{I}}$ is then learned for transferring the information prior from image-level to text instance-level, which is defined as:

$$\tilde{\boldsymbol{I}} = \mathrm{TDec}(Q = \boldsymbol{I}, K = \boldsymbol{t}_{\text{out}}, V = \boldsymbol{t}_{\text{out}}) \in \mathbb{R}^{\tilde{H}\times\tilde{W}\times C}, \tag{9}$$

where TDec denotes the Transformer Decoder. In practice, it consists of 6 bidirectional transformer decoder layers with 4 heads for adequately interacting between image embeddings and text embeddings; transformer width is 256, and the feed-forward hidden dimension is set to 1024.

Based on the conditional visual prompt, the original image embedding $\boldsymbol{I}$ is equipped with $\tilde{\boldsymbol{I}}$ to produce the prompted text-aware locality embeddings $\hat{\boldsymbol{I}}$ used for instance-language matching (11) and downstream detection and spotting head:

$$\hat{\boldsymbol{I}} = \boldsymbol{I} + \tilde{\boldsymbol{I}}. \tag{10}$$

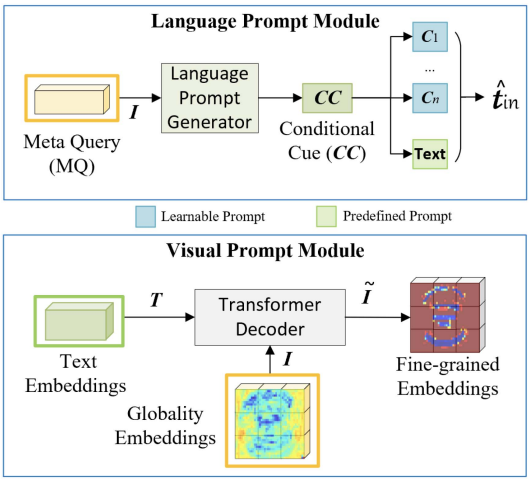


Fig. 5. Illustration of the language prompt module (top) and visual prompt module (bottom).

5) Instance-Language Matching: Given the output of the text encoder and image encoder, we perform text instance-language matching alignment on text-aware locality image embedding $\hat{\boldsymbol{I}}$ and text embedding $\boldsymbol{t}_{\text{out}}$ by the dot product followed by sigmoid activation to get binary score map. The mixture of the generated conditional fine-grained embedding $\tilde{\boldsymbol{I}}$ and visual embedding $\boldsymbol{I}$ can allow text instances existing in visual features to be better matched with pretrained language knowledge in collaboration. The matching mechanism is formulated as follows:

$$\boldsymbol{P} = \text{sigmoid}(\hat{\boldsymbol{I}}\boldsymbol{t}_{\text{out}}^{T}/\tau) \in \mathbb{R}^{\tilde{H}\times\tilde{W}\times 1}, \tag{11}$$

where $\boldsymbol{t}_{\text{out}}$ is text embedding because of only one text class in text detection scenarios, and τ is the temperature coefficient which is empirically set to 0.07 by default. $\boldsymbol{P}$ is the binary text segmentation map. The segmentation maps are supervised using the ground-truths as an auxiliary loss and concatenated by the prompted embedding $\hat{\boldsymbol{I}}$ for downstream text detection and spotting head to explicitly incorporate language priors for detection. During training, we minimize a binary cross-entropy loss between the segmentation map $\boldsymbol{P}$ and ground-truth, which is defined as follows:

$$\mathcal{L}_{aux} = \sum_{i}^{\tilde{H}}\sum_{j}^{\tilde{W}} y_{ij}\log(P_{ij}) + (1-y_{ij})\log(1-P_{ij}), \tag{12}$$

where y_{ij} and P_{ij} are the label and predicted probability of pixel (i, j) belonging to the text instances, respectively.

C. Optimization

The loss function $\mathcal{L}_{total}$ is the sum of task loss $\mathcal{L}_{task}$ and auxiliary loss $\mathcal{L}_{aux}$, formulated as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \lambda\mathcal{L}_{aux}, \tag{13}$$

where λ is a trade-off hyper-parameters and set to 1 in this paper. $\mathcal{L}_{task}$ depends on downstream text detection methods including segmentation and regression categories, or text spotting methods. In the inference period, we use the output of the corresponding task head as the final result. In practice, we integrate the proposed method into both text detectors and text spotters to validate the effectiveness of our methods.

IV. Experiments

We conduct extensive experiments to validate FastTCM. Our first set of experiments examines how FastTCM-CR50 backbone can be incorporated into existing text detectors and spotters to achieve consistent performance improvements. Next, we demonstrate the few-shot training capability and generalization ability by incorporating the FastTCM method. In the third set of experiments, we compare our method with previous pretraining methods tailored for text detection and spotting. Then, we provide thorough experiments to evaluate the sensitivity w.r.t. the proposed designs. Finally, we also conducted experiments on challenging oriented aerial object detection datasets to demonstrate the effectiveness of our method.

A. Datasets

Our experiments are conducted on a number of commonly known scene text detection and spotting benchmarks including ICDAR2013 (IC13) [58], ICDAR2015 (IC15) [59], MSRA-TD500 (TD) [60], CTW1500 (CTW) [61], Total-Text (TT) [62], ArT [63], MLT17 [64], MLT19 [65], SynthText [66], CurvedSynthText-150k [44], and TextOCR [67]. More details of the datasets refer to appendix, available online.

B. Implementation Details

In our text detection task experiments, we test the efficacy of prominent detection methodologies including DBNet (DB) [14], PAN [13], FCENet (FCE) [25], and a newer methods TextPMs [68]. The detection head from DBNet, PAN, FCENet, and TextPMs are utilized to yield the final results. To test the model's few-shot learning, we train on the benchmark using varying proportions of training data, and evaluate it against the corresponding test data. The generalization capability is tested by training it on respective source datasets and subsequently evaluating it on a target dataset with a markedly different distribution. The generalization ablity of the FastTCM-CR50 is assessed through two different forms of adaptation: synthtext-to-real and real-to-real. A series of ablation studies are undertaken, focusing on the predefined prompt, the learnable prompt, the language prompt module, the visual prompt generator, the BSM module, and various settings.

For end-to-end text spotting tasks, we carry out experiments with recent methods such as Mask TextSpotter v3 (MTSv3) [43], ABCNet [44], ABINet++ [45], TESTR [49], and DeepSolo [52]. These selected methods encompass both RoI-based and RoI-free text spotters. To ensure consistency with these text spotting methods, we use the same training approach, respecting the training data and hyper-parameters specific to each method.

C. Cooperation With Existing Detector Methods

We assessed the impact of substituting the original backbones (ResNet50) of FCENet, PAN, DBNet, and TextPMs with the pretrained image encoder ResNet50 from CLIP (CR50). Yet, as evidenced in Table I, merely leveraging the pretrained

TABLE I
TEXT DETECTION RESULTS OF COOPERATING WITH EXISTING DETECTORS ON IC15, TD, AND CTW

Method	BB	IC15		TD		CTW		FPS
		F	Δ	F	Δ	F	Δ	
TextSnake [11]	R50	82.6	-	78.3	-	75.6	-	1.1
TextField [70]	VGG16	84.1	-	81.3	-	81.4	-	-
PSENet-1s [12]	R50	85.7	-	-	-	82.2	-	1.6
LOMO [23]	R50	87.2	-	-	-	78.4	-	-
CRAFT [71]	VGG16	86.9	-	82.9	-	83.5	-	8.6
ContourNet [72]	R50	86.9	-	-	-	83.9	-	-
DRRG [73]	VGG16	86.6	-	85.1	-	84.5	-	-
MOST [26]	R50	88.2	-	86.4	-	-	-	10.1
Raisi *et al.* [74]	R50	83.7	-	87.2	-	-	-	-
Tang *et al.* [15]	R50	90.9	-	87.3	-	89.1	-	-
TextFuseNet [75]	R50	90.1	-	-	-	85.4	-	-
DB++ [76]	R50	87.3	-	87.2	-	85.3	-	27
TextBPN [27]	R50	-	-	85.6	-	85.0	-	-
FCE [25]	R50	86.2	-	85.4†	-	85.5	-	11.5
	CR50	86.2	**+0.2**	86.1	**+0.7**	85.5	**+0.1**	11.5
	TCM-CR50	87.1	**+0.9**	86.9	**+1.5**	85.9	**+0.4**	8.4
	FastTCM-CR50	87.3	**+1.1**	87.1	**+1.7**	86.0	**+0.5**	10.3
PAN [13]	R50	82.9	-	84.1	-	83.7	-	36
	CR50	83.2	**+0.3**	84.6	**+0.5**	83.9	**+0.2**	36
	TCM-CR50	84.6	**+1.7**	85.3	**+1.2**	84.3	**+0.6**	18
	FastTCM-CR50	84.9	**+2.0**	85.4	**+1.3**	84.5	**+0.8**	31
DB [14]	R50	87.3	-	84.9	-	83.4	-	14.5
	CR50	87.7	**+0.4**	86.8	**+1.9**	83.4	**+0**	14.5
	TCM-CR50	89.2	**+1.9**	88.8	**+3.9**	84.9	**+1.5**	10
	FastTCM-CR50	89.4	**+2.1**	88.9	**+4.0**	85.2	**+1.8**	13.3
TextPMs [68]	R50	87.3	-	88.9	-	85.7	-	10.5
	CR50	87.6	**+0.3**	88.9	**+0**	85.8	**+0.1**	10.5
	TCM-CR50	89.6	**+2.3**	89.1	**+0.2**	86.0	**+0.3**	7.2
	FastTCM-CR50	90.0	**+2.7**	89.3	**+0.4**	86.2	**+0.5**	9.1

† indicates the results from [69]. "BB" denotes backbone where R50, CR50, TCM-CR50, FastTCM-CR50 Represent the original ResNet50 backbones, the pretrained clip ResNet50 backbone, the TCM-CR50 backbone, and ours FastTCM-CR50 backbone respectively. F (%) represents F-measure. Δ means the improvement of performance between the cooperated method and the original method. FPS are reported using a single V100
The bold font stand for the performance variance.

visual-language knowledge of the CLIP model (CR50) is inadequate for boosting scene text detection performance. This suggests the necessity of employing an appropriate method to harness the potential of the CLIP model. Subsequently, we evaluated the performance of FastTCM-CR50 with these two backbones. As illustrated in Table I, FastTCM-CR50 can be effectively employed to augment current scene text detectors, yielding an average improvement of 1.6% compared to the respective baseline methods. Furthermore, it is demonstrated that the FastTCM-CR50 backbone surpasses the TCM-CR50 backbone in terms of F-measure, contributing to an average performance enhancement of 0.2% for DBNet, FCENet, PAN, and TextPMs on the IC15, TD, and CTW datasets, with an average speed improvement of 45.87%. Furthermore, with the incorporation of stronger detectors like TextPMs, FastTCM consistently delivered an average performance increase of 1.2% across various text detection datasets compared to baseline method. This improvement demonstrates the adaptability of our method, further establishing its benefits even when integrated with top-tier detection algorithms.

We visualize our method in Fig. 6. It shows that the fine-grained features $\tilde{\boldsymbol{I}}$ containing text information is recovered from the global image embedding $\boldsymbol{I}$, demonstrating that FastTCM can identify text regions and provide these prior cues for downstream text perception related tasks.

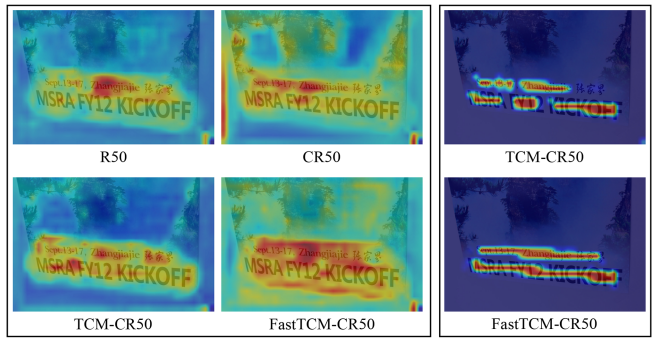


Fig. 6. Visualization results of our method. The left is the image embedding $\boldsymbol{I}$ of different backbone including R50, CR50, TCM-CR50, and FastTCM-CR50, and the right is the generated visual prompt $\tilde{\boldsymbol{I}}$. Our method FastTCM-CR50 can accurately identify text regions. Best view in screen.

D. Cooperation With Existing Spotter Methods

Detection-only Results: As demonstrated in Table II, we noted consistent enhancements in F-measure on text spotting benchmarks when TCM-CR50 was combined with five distinct text spotting methods. Particularly, TCM-CR50 outperformed the baseline methods such as MTSv3, ABINet++, ABCNet, DeepSolo, and TESTR with an R50 backbone, with performance boosts ranging from +0.2% to +1.8% in terms of F-measure on the TT dataset. Consistent improvements were also witnessed on IC15 and CTW datasets, underscoring TCM-CR50's suitability for text-spotting methods. Furthermore, when FastTCM-CR50 was integrated with MTSv3, ABINet++, ABCNet, DeepSolo, and TESTR, we observed an average performance enhancement of 0.2% compared to TCM-CR50 based methods, accompanied by similar speed improvements, indicating FastTCM-CR50's superior efficacy. Additionally, the inclusion of an extra large-scale dataset, TextOCR, resulted in further performance gains, such as a 0.9% improvement on the TT dataset using TESTR.

End-to-end Spotting Results: In Table II, we present the end-to-end spotting performance of our method combined with existing scene text spotters. TCM-CR50 demonstrates favorable performance when integrated with various cooperative methods. Specifically, under the end-to-end setting with the strong lexicon on dataset IC15, TCM-CR50 outperforms the original MTSv3, ABINet++, ABCNet, DeepSolo, and TESTR by +0.8%, +0.3%, +2.3%, +0.1%, and +0.4%, respectively, in terms of the 'S' metric. Similar consistent improvements are also observed for datasets TT and CTW, indicating that TCM-CR50 effectively enhances the performance of both existing scene text detectors and spotters. Furthermore, when replacing TCM-CR50 with FastTCM-CF50, we observe a further improvement in performance, with an average gain of 1.5% compared to baseline methods and an average gain of 0.55% compared to TCM-CR50. Additionally, the inference speed of FastTCM-CR50 is increased by approximately 46.4%. These results highlight the superiority of FastTCM-CR50 and its potential for efficient

TABLE II
END-TO-END TEXT SPOTTING RESULTS OF COOPERATING WITH EXISTING SPOTTER METHODS ON TOTAL-TEXT, ICDAR2015, AND CTW1500

Method	BB	Ext.	Detection Results						End-to-End Results											FPS
			TT		IC15		CTW		TT			IC15				CTW				
			F	Δ	F	Δ	F	Δ	None	Full	Δ (None)	S	W	G	Δ (S)	None	Full	Δ (None)	
CharNet [77]	R50	-	84.6	-	89.7	-	-	-	48.8	78.8	-	80.1	74.5	62.2	-	-	-	-	5.4
CRAFTS [78]	R50	-	87.4	-	87.1	-	-	-	78.8	-	-	83.1	82.1	74.9	-	-	-	-	8.8
TextPerceptron [79]	R50	-	85.2	-	87.1	-	84.6	-	69.7	78.3	-	80.1	76.6	65.1	-	57.0	-	-	-
TextDragon [80]	VGG16	-	80.3	-	87.8	-	83.6	-	48.8	74.8	-	82.5	78.3	65.2	-	39.7	72.4	-	2.6
Boundary [41]	R50	-	87.0	-	88.6	-	-	-	65.0	76.1	-	79.7	75.2	64.1	-	-	-	-	-
PAN++ [81]	R18	-	-	-	87.5	-	84.0	-	68.6	78.6	-	82.7	78.2	69.2	-	-	-	-	36.0
PGNet [82]	R50	-	86.1	-	88.2	-	-	-	63.1	-	-	88.3	78.3	63.5	-	-	-	-	38.2
MANGO [83]	R50	-	-	-	-	-	-	-	72.9	83.6	-	85.4	80.1	73.9	-	58.9	78.7	-	8.4
GLASS [84]	R50	-	88.1	-	85.7	-	-	-	79.9	86.2	-	84.7	80.1	76.3	-	-	-	-	2.7
MTSv2 [38]	R50	-	78.5	-	87.0	-	-	-	65.3	77.4	-	83.0	77.7	73.5	-	-	-	-	3.1
ABCNetv2 [85]	R50	-	87.0	-	88.1	-	84.7	-	73.5	80.7	-	83.0	80.7	75.0	-	58.4	79.0	-	10
SwinTS [46]	Swin-T	-	88.0	-	-	-	88.0	-	74.3	84.1	-	83.9	77.3	70.5	-	51.8	77.0	-	2.9
SPTS [51]	R50	-	-	-	-	-	-	-	74.2	82.4	-	77.5	70.2	65.8	-	63.6	83.8	-	0.6
TTS [50]	R50	-	-	-	-	-	-	-	78.2	86.3	-	85.2	81.7	77.4	-	-	-	-	-
MTSv3 [43]	R50	-	79.7	-	87.5	-	83.7	-	71.2	78.4	-	83.3	78.1	74.2	-	52.6	75.8	-	2.5
	oCLIP-R50	-	80.0	**+0.3**	87.7	**+0.2**	83.9	**+0.2**	71.8	79.6	**+0.6**	83.8	78.6	74.3	**+0.5**	53.0	76.4	**+0.4**	2.5
	TCM-CR50	-	81.5	**+1.8**	88.0	**+0.5**	84.1	**+0.4**	73.7	82.6	**+2.5**	84.1	80.2	75.5	**+0.8**	53.2	76.7	**+0.6**	1.3
	FastTCM-CR50	-	81.9	**+2.2**	88.2	**+0.7**	84.2	**+0.5**	73.9	83.0	**+2.7**	84.5	80.3	76.0	**+1.2**	54.0	77.2	**+1.4**	1.9
ABINet++ [45]	R50	-	86.0	-	88.2	-	84.0	-	77.6	84.5	-	84.1	80.4	75.4	-	60.2	80.3	-	10.6
	TCM-CR50	-	86.3	**+0.3**	88.4	**+0.2**	84.3	**+0.3**	77.7	84.8	**+0.1**	84.4	80.7	75.7	**+0.3**	60.4	80.5	**+0.2**	5.1
	FastTCM-CR50	-	86.5	**+0.5**	88.8	**+0.6**	84.7	**+0.7**	77.9	85.0	**+0.3**	85.1	80.9	76.0	**+1.0**	60.5	80.8	**+0.3**	9.7
ABCNet [44]	R50	-	86.0	-	86.8	-	84.4	-	64.2	75.7	-	79.2	74.1	66.8	-	45.2	74.1	-	17.9
	CR50	-	86.1	**+0.1**	87.2	**+0.4**	84.5	**+0.1**	64.3	76.4	**+0.1**	79.4	75.0	67.3	**+0.2**	46.1	75.3	**+0.9**	17.9
	TCM-CR50	-	86.4	**+0.4**	88.4	**+1.6**	84.8	**+0.4**	68.3	81.8	**+4.1**	81.5	77.2	72.3	**+2.3**	48.2	74.6	**+3.0**	16.4
	FastTCM-CR50	-	86.6	**+0.6**	88.9	**+2.1**	85.3	**+0.9**	69.5	82.4	**+5.3**	82.8	78.1	72.8	**+3.6**	49.1	77.8	**+3.9**	17.2
DeepSolo [52]	R50	-	87.3	-	90.0	-	87.2	-	79.7	87.0	-	86.8	81.9	76.9	-	60.1	78.4	-	17.0
	CR50	-	87.4	**+0.1**	90.1	**+0.1**	87.2	**+0**	79.7	87.0	**+0**	86.8	81.9	77.0	**+0**	60.2	78.4	**+0.1**	17.0
	TCM-CR50	-	87.5	**+0.2**	90.2	**+0.2**	87.3	**+0.1**	79.8	87.1	**+0.1**	86.9	82.0	77.1	**+0.1**	60.3	78.5	**+0.2**	8.3
	FastTCM-CR50	-	87.8	**+0.5**	90.3	**+0.3**	87.4	**+0.2**	79.9	87.2	**+0.2**	87.0	82.0	77.3	**+0.2**	60.4	78.8	**+0.3**	15.1
TESTR [49] (polygon)	R50	-	86.9	-	90.0	-	87.1	-	73.2	83.9	-	85.2	79.4	73.6	-	55.9	81.5	-	12.1
	CR50	-	87.1	**+0.2**	90.0	**+0**	87.1	**+0**	73.3	84.0	**+0.1**	85.4	80.6	74.6	**+0.3**	56.1	81.6	**+0.2**	12.1
	TCM-CR50	-	87.9	**+1.0**	90.1	**+0.1**	87.2	**+0.1**	73.6	84.1	**+0.4**	85.6	80.3	74.2	**+0.4**	56.2	81.8	**+0.3**	6.2
	FastTCM-CR50	-	88.1	**+1.2**	90.2	**+0.2**	87.3	**+0.2**	74.2	85.4	**+1.0**	86.2	80.9	75.2	**+1.0**	56.8	82.4	**+0.9**	10.7
	FastTCM-CR50	✓	89.0	**+2.1**	90.3	**+0.3**	87.4	**+0.3**	76.5	86.9	**+3.3**	86.8	81.6	76.1	**+1.6**	57.5	82.8	**+1.6**	10.7

"Ext." short for extra data TextOCR. "S", "W", and "G" donate using strong, weak, and generic lexicons, respectively. F (%) represents F-measure. Δ means the improvement of performance between the cooperated method and the original method.
The bold font stand for the performance variance.

and accurate text spotting tasks. Besides, when using additional large-scale TextOCR as training data, our model can achieve further improvement, suggesting the compatibility of our method with large-scale datasets.

E. Few-Shot Training Ability

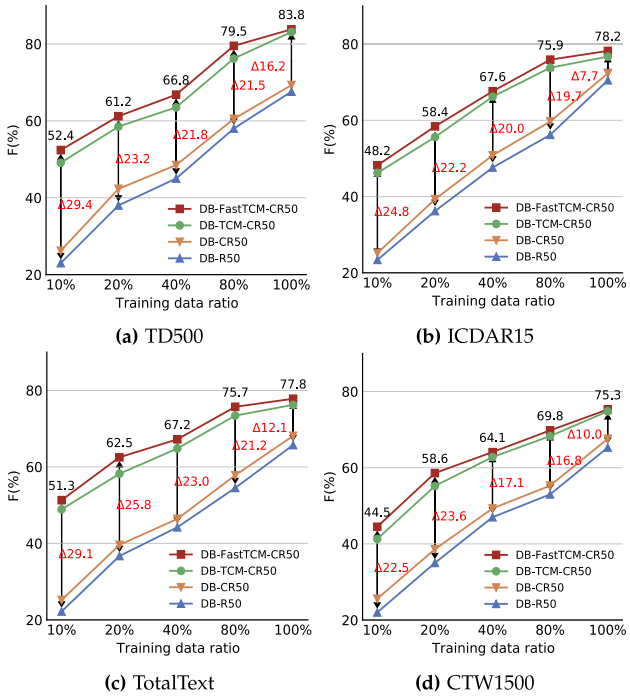
Results for Text Detection Task: To verify the few-show training ability of our method on text detection tasks, we directly train our model on real datasets using various training data ratios without pretraining, and evaluate it on the corresponding 4 benchmarks. As shown in Fig. 7, DB-FastTCM-CR50 shows robustness on limited data and outperforms the baseline methods DB in an average of 26.5% in terms of 10% training data ratio settings. Besides, DB-CR50 has limited improvements compared to our specific design FastTCM. The results show that the FastTCM can capture the inherent characteristic of text via leveraging the pretrained vision and language knowledge of the zero-shot trained CLIP model.

Few-shot Experiments for Text Spotting: In addition, we performed few-shot experiments on text spotting tasks using ABCNet, TESTR, and DeepSolo on Total-Text, as illustrated in Table III. Considering that the recognizer module in text spotting methods often struggles to learn effectively with very limited data, we followed the text spotting pretraining step to obtain a suitable initialization for the corresponding text spotting methods. Subsequently, we applied different training

TABLE III
FEW-SHOT TRAINING ABILITY OF TEXT SPOTTING TASK WITH VARYING TRAINING DATA RATIO ON TOTAL-TEXT

Method	BB	10%	20%	40%	80%	100%
ABCNet	R50	43.5	51.7	55.2	62.3	64.2
	CR50	46.2	53.5	58.6	63.7	64.3
	TCM-CR50	50.6	57.2	63.2	67.3	68.3
	FastTCM-CR50	**52.3**	**60.1**	**66.2**	**68.0**	**69.5**
TESTR	R50	58.4	62.2	68.5	69.3	73.2
	CR50	59.1	63.9	68.7	70.4	73.3
	TCM-CR50	60.3	66.8	68.9	72.8	73.6
	FastTCM-CR50	**60.5**	**67.0**	**69.2**	**73.4**	**74.2**
DeepSolo	R50	61.2	64.6	70.1	73.5	79.7
	CR50	61.5	65.2	70.8	74.8	79.7
	TCM-CR50	63.6	68.7	73.2	76.5	79.8
	FastTCM-CR50	**64.3**	**69.1**	**74.3**	**77.6**	**79.9**

End-to-End spotting metric "None" (%) is reported.
The bold font stand for the best performance.



(a) TD500 **(b)** ICDAR15 **(c)** TotalText **(d)** CTW1500

Fig. 7. Few-shot training ability of text detection task with varying training data ratio. "F" represents F-measure.

ratios of the Total-Text dataset to evaluate the few-shot learning ability. The results demonstrate that both TCM-CR50 and FastTCM-CR50 exhibit advantages in few-shot learning for text spotting tasks compared to DB-R50 and DB-CR50. Moreover, using our method outperforms baseline methods by an average of 4.7%. This demonstrates the effectiveness and superiority of FastTCM-CR50 over simply replacing other counterparts. Furthermore, as shown in DeepSolo item of Table III, the results show that FastTCM facilitates an average performance improvement of 3.2% over the training-efficient method DeepSolo. This enhancement demonstrates FastTCM's compatibility with training-efficient methods and its significant advantage in few-shot learning scenarios for text spotting tasks, where it broadens the applicability and utility of our approach in real-world scenarios.

F. Generalization Ability

CLIP Backbone Generalization: We conducted an experiment to investigate the generalization performance of DBNet by directly replacing the backbone of DBNet with CLIP backbone (CR50), as shown in Table IV. It shows that the CLIP-R50 can indeed bring benefits for generalization. However, by integrating with FastTCM-CR50 backbone, the performance can be significantly improved. It suggests that directly using the pretrained CLIP-R50 is not strong enough to improve the generalization performance of the existing text detector, which further indicates that synergistic interaction between the detector and the CLIP is important. Meanwhile, FastTCM-CR50 also consistently outperforms TCM-CR50.

Synth-to-real and real-to-real Adaptation: We conduct two types of experiments including synthtext-to-real adaptation and real-to-real adaptation on text detection tasks, as shown in Tables IV and V, respectively. Real-to-real adaptation contains monolingual and multi-lingual scenarios. From the tables, we can see that by integrating the FastTCM-CR50 into DBNet, we significantly improve the performance by an average of 12.4% in terms of F-measure for four different settings including synthtext-to-real and real-to-real, which further demonstrates the effectiveness of our method for domain adaptation. Notably, FastTCM-CR50 also consistently demonstrates improvements by an average of 0.4% compared to TCM-CR50 as well, further emphasizing the remarkable generalization ablity of our methods.

Real-to-Real Adaptation on Scene Text Spotting: Besides, we also conducted real-to-real adaptation experiments with existing spotting methods, as shown in Table VI. The results show that the FastTCM-CR50 has the capacity of improving the existing scene text spotting methods by an average of 14.8%, further demonstrating the effective generalization ability.

TABLE IV
SYNTHTEXT-TO-REAL ADAPTATION

Method	BB	ST → IC13	ST → IC15
EAST† [22]	PVANet [87]	67.1	60.5
PAN [13]	R50	-	54.8
CCN [77]	R50	-	65.1
ST3D [88]	R50	73.8	67.6
DB [14]	R50	71.7	64.0
	CR50	73.1	67.4
	TCM-CR50	79.6	76.7
	FastTCM-CR50	**79.9**	**77.2**

†indicates the results from [86]. ST indicates SynthText. F-measure (%) is reported.
The bold font stand for the best performance.

TABLE V
REAL-TO-REAL ADAPTATION

Method	BB	IC13→IC15	IC13→TD	MLT17→MLT19
EAST† [22]	PVANet	53.3	46.8	-
GD(AD) [69]	-	64.4	58.5	-
GD(10-AD) [69]	-	69.4	62.1	-
CycleGAN [89]	-	57.2	-	-
ST-GAN [90]	-	57.6	-	-
TST [86]	PVANet	52.4	-	-
DB [14]	R50	63.9	53.8	47.4
	TCM-CR50	71.9	65.1	67.5
	FastTCM-CR50	**72.4**	**65.7**	**67.8**

†indicates that the results are from [69]. Note that the proposed method outperforms other methods. F-measure (%) is reported.
The bold font stand for the best performance.

TABLE VI
REAL-TO-REAL ADAPTATION ON SCENE TEXT SPOTTING METHODS
END-TO-END SPOTTING METRIC "NONE" (%) IS REPORTED

Method	BB	TT→IC15	TT→CTW	IC15→CTW	CTW→IC15
ABCNet	R50	32.5	37.1	33.5	34.6
	FastTCM-CR50	**48.2**	**50.3**	**47.8**	**51.7**
TESTR	R50	36.7	39.2	36.1	36.8
	FastTCM-CR50	**53.1**	**49.2**	**48.1**	**56.2**

The bold font stand for the best performance.

TABLE VII
COMPARISON WITH EXISTING SCENE TEXT PRETEXT TASK PRETRAINING TECHNIQUES ON DBNET (DB)

	Method	BB	IC15	TT	TD	CTW
Convention	SegLink [10]	VGG16	-	-	77.0	-
	PSENet-1s [12]	R50	85.7	80.9	-	82.2
	LOMO [23]	R50	87.2	81.6	-	78.4
	MOST [26]	R50	88.2	-	86.4	-
	Tang *et al.* [15]	R50	89.1	-	88.1	-
VLP	DB+ST†	R50	85.4	84.7	84.9	-
	DB+STKM† [7]	R50	86.1	85.5	85.9	-
	DB+VLPT† [6]	R50	86.5	86.3	88.5	-
	DB+oCLIP* [8]	R50	85.4	84.1	-	82.0
	DB	TCM-CR50	89.4	85.9	88.8	85.1
	DB	FastTCM-CR50	**89.5**	86.1	**88.9**	**85.2**
	DB+oCLIP* [8]	FastTCM-CR50	**89.6**	86.2	**88.94**	**85.4**

†indicates the results from [6]. ST and VLP denote SynthText pretraining and visuallanguage pretext task pretraining methods, respectively. * stand for the results from [8]. F-measure (%) is reported.
The bold font stand for the best performance.

TABLE VIII
ABLATION STUDY OF OUR PROPOSED COMPONENTS ON IC15, TD, TT, AND CTW

Method	PP	LP	LM	VG	Aux.	BSM	IC15 F	TD F	TT F	CTW F
BSL	×	×	×	×	✓	×	87.7	86.8	84.7	83.4
BSL+	✓	×	×	×	✓	×	87.75	87.0	84.74	83.5
	✓	4	×	×	✓	×	88.0	87.1	84.8	83.6
	×	4	×	×	✓	×	87.8	87.7	85.1	83.9
	×	18	×	×	✓	×	88.1	87.8	85.3	83.9
	×	32	×	×	✓	×	88.4	88.2	85.4	84.5
	✓	4	✓	×	✓	×	88.6	88.4	85.5	84.6
	✓	4	✓	✓	✓	×	89.2	88.9	85.6	84.9
	✓	32	✓	✓	✓	×	89.4	88.8	85.9	85.1
	✓	4	✓	✓	×	✓	87.9	87.2	84.6	84.2
FastTCM-CR50	✓	4	✓	✓	✓	✓	**89.4**	**88.9**	**86.1**	**85.2**

"BSL", "PP", "LP", "LM", "VG", "Aux.", and "BSM" represent the baseline method DBNet, the predefined prompt, the learnable prompt, the language prompt module, the visual prompt generator, auxiliary loss, and bimodal similarity matching, respectively. F (%) represents F-measure. Δ represents the variance.
The bold font stand for the best performance.

G. Comparison With Pretraining Methods

The pretraining methods based on specifically designed pretext tasks have made effective progress in the field of text detection. In contrast to these efforts, FastTCM-CR50 can turn the CLIP model directly into a scene text detector without pretext task pretraining process. The comparison results are shown in Table VII, from which we can see that without pretext tasks for pretraining, DB+FastTCM-CR50 consistently outperforms previous methods including DB+STKM [7], DB+VLPT [6], and DB+oCLIP [8]. Especially on IC15, our method outperforms the previous state-of-the-art pretext task pretraining method by a large margin, with 89.5% versus 86.5% in terms of the F-measure. Furthermore, we demonstrate the proposed backbone can also be further improved using such pretext tasks pretraining as in oCLIP, with an average of 0.11% improvement in terms of the F-measure.

H. Ablation Studies

Ablation Study for the Predefined Prompt: When using the predefined prompt, as illustrated in the second row of Table VIII, the performances are slightly improved on all four datasets (IC15, TD, TT, and CTW), with 0.05%, 0.2%, 0.04%, and 0.1% higher than the baseline method, respectively.

Ablation Study for the Learnable Prompt: Then, results combing the learnable prompt with the predefined prompt on four datasets are provided in the third row of Table VIII. We notice that a consistent improvement can be achieved by adding the learnable prompt. We also show the influence of using different numbers of the learnable prompt in row 4 to row 6 of Table VIII. We observe that as the value of the number of the learnable prompt increases, the performance increases gradually on all datasets. Compared to the value 4, the value 32 obtains obvious improvements on CTW, TD, and TT. We conjecture that this is because the larger number of the learnable prompt can better steer the pretrained text encoder knowledge which is useful for text detection. In the following experiments, the default number of the learnable prompt is set to 4 for simplicity.

Ablation Study for the Language Prompt Module: Besides, we evaluate the performance of the proposed language prompt module shown in 7_{th} row of Table VIII. With the help of the language prompt module, we find that TCM achieves further improvements on all four datasets, especially on ICDAR2015, indicating that the conditional cue generated by the language prompt module for each image can ensure better generalization over different types of datasets.

Ablation Study for the Visual Prompt Generator: Furthermore, combining the proposed visual prompt generator with the above other components, the improvement of F-measure is better than the baseline on all four datasets, with larger margins of 1.7% and 2.0% on IC15 and TD, respectively. The reason for this obvious complementary phenomenon is that the visual prompt generator can propagate fine-grained visual semantic information from textual features to visual features. Besides, the prompted locality image embedding generated by the visual prompt generator can guide the model to obtain more accurate text instance-level visual representations, which boosts the ability of instance-language matching and generates a precise segmentation score map that is useful for downstream detection head.

Ablation Study for the Bimodal Similarity Matching: We further conducted a comparison of the results with and without bimodal similarity matching, as outlined in the 7th row of the BSL+ group of Table VIII. The results clearly demonstrate that the utilization of bimodal similarity matching leads to higher performance. This finding indicates that bimodal similarity matching plays a crucial role in training the model by dynamically enriching text embeddings with visual information, resulting in improved overall performance.

TABLE IX
ABLATION STUDY OF THE EFFECT OF META QUERY, BSM, LM, AND VG ON GENERALIZATION PERFORMANCE

Method	BB	TT → IC15	TT → CTW	IC15 → CTW
TESTR	FastTCM-CR50	**53.1**	**49.2**	**48.1**
TESTR	w/o MQ	50.4 (–2.7)	45.5 (–3.7)	44.6 (3.5)
	w/o BSM	48.2 (–3.9)	43.4 (–5.8)	42.9 (–5.2)
	w/o LM	45.5 (–7.6)	40.3 (–9.0)	41.7 (–6.4)
	w/o VG	46.6 (–6.5)	42.4 (–6.8)	42.0 (–6.1)
	w/o All	41.4 (–11.7)	39.2 (–10)	37.3 (–10.8)

MQ is short for meta query. F-measure (%) is reported.
The bold font stand for the best performance.

TABLE X
ABLATION STUDY OF THE TRAINABLE PARAMETERS COMPARISON WITH DBNET ON TD DATASET AND IC13 → IC15

Method	BB	Num.	Params	FLOPs	TD	IC13 → IC15	FPS
DB	R50	-	26 (M)	98 (G)	84.9	63.9	14.5
	R101	-	46 (M)	139 (G)	85.9	64.3	11.7
	R152	-	62 (M)	180 (G)	87.3	64.7	8.4
DB	TCM-CR50	6	50 (M)	156 (G)	88.7	71.9	10
	FastTCM-CR50	1	30 (M)	107 (G)	88.1	72.0	14.2
	FastTCM-CR50	3	34 (M)	117 (G)	88.5	72.2	13.8
	FastTCM-CR50	6	50 (M)	156 (G)	**88.9**	**72.4**	13.3

"Num." refer to the number of transformer decoder layers of VG. F-measure (%) is reported.
The bold font stand for the best performance.

TABLE XI
ABLATION STUDY OF THE DIFFERENT PREDEFINED LANGUAGE PROMPT WITH DBNET-FASTTCM-CR50 ON TD

Predefined language prompt	TD
"Text"	**88.9**
"A set of arbitrary-shape text instances"	88.7
"The pixels of many arbitrary-shape text instances"	88.6
w/o predefined language prompt	87.4

F-measure (%) is reported.
The bold font stand for the best performance.

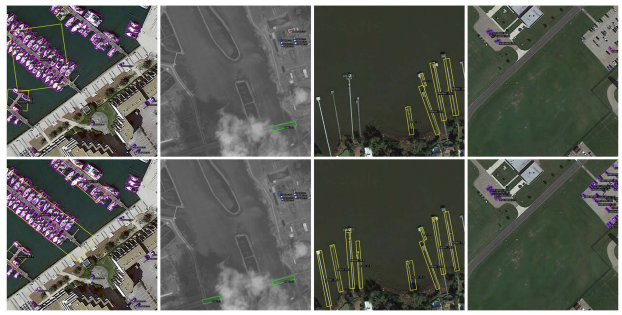


Fig. 8. Top row and bottom row are qualitative results on DOTA-v1.0 testing set without and with cooperating with FastTCM-CR50, respectively. It contains 15 common categories, such as ship, small-vehicle, harbor, bridge, basketball-court, storage-tank, etc.

Ablation Study for the Auxiliary Loss: We compare the results of with and without auxiliary loss, as shown in the last row of the BSL+ group of Table VIII. We observe that using auxiliary loss achieves higher performance. The results indicate auxiliary loss is beneficial to train the model via imposing constraints on instance-language matching score map. In addition, the improvement of the performance suggests that it might help the image encoder of pretrained CLIP to perceive locality text regions effectively.

Ablation Study for the Key Component on Generalization Performance: As presented in Table IX, removing the meta query and BSM elements from FastTCM dramatically deteriorates the generalization performance, highlighting the importance and effectiveness of these components. Similarly, removing the VG and LM elements from FastTCM also results in a substantial drop in generalization performance, further validating their effectiveness. Finally, when we remove all of these components, the performance experiences an additional significant drop, indicating that each of these components contributes to the overall effectiveness and performance of FastTCM-CR50.

Ablation Study for the Parameters Comparison: For a fair comparison, we have increased the parameters of DBNet by replacing the backbone with a larger ResNet and then conducting text detection experiments on TD dataset and a domain adaptation experiment on IC13 → IC15. Trainable parameters and FLOPs are calculated with an input size of 1280 × 800. Results are shown in Table X. The results show that DBNet with FastTCM-CR50 has better performance than DBNet with less model size and computation overhead compared to DBNet with R152 backbone, demonstrating its effectiveness.

Ablation Study for the number of transformer decoder layers of VG: The last group of Table X demonstrates the impact of varying the number of transformer decoder layers of VG on the performance. The results show that the performance remains robust across different numbers of decoder layers. This indicates that in practical applications, we have the flexibility to decrease the number of transformer decoder layers to achieve a better trade-off between model parameters and performance. By reducing the number of layers, we can potentially save computational resources and memory while maintaining satisfactory performance, making the model more efficient and practical for real-world applications.

Ablation Study for the Different Predefined Language Prompt: We conducted ablation study on the predefined language prompt with different strings using DBNet with FastTCM-CR50 in Table XI. Results show that without predefined language prompt, the performance is harmed. In addition, it can be seen that there is little performance variation with different predefined language prompt. When the predefined language prompt becomes long and complex, the model performance drops a little. We deem that the CLIP is not good at handling complex instructions because it is pretrained on a dataset of 400 million image-text pairs that contain noise. As a result, this noise can affect the CLIP's ability to deal with long instructions.

Ablation Study for Different Amount of Data: To further explore whether the FastTCM can learn the additional knowledge which is hard to be obtained from increasing data, we have trained the model on large-scale public joint data including IC13, IC15, TD, CTW, TT, and MLT17, with a total of 13,784 images, and testing it on a NightTime-ArT data (326 images) carefully collected from ArT. The nighttime examples of ArT

TABLE XII
DETECTION RESULTS OF COOPERATING WITH EXISTING ROTATED OBJECT DETECTION METHODS ON THE DOTA-V1.0 TESTING SET

	Method	BB	MS	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP_{50}
Single-stage	PolarDet [91]	R101	√	89.7	87.1	48.1	71.0	78.5	80.3	87.5	90.8	85.6	86.9	61.6	70.3	71.9	73.1	67.1	76.6
	RDD [92]	R101	√	89.1	83.9	52.5	73.1	77.8	79.0	87.1	90.6	86.7	87.1	64.1	70.3	77.0	75.8	72.2	77.8
	GWD [93]	R152	√	89.1	84.3	55.3	77.5	77.0	70.3	84.0	89.8	84.5	86.1	73.5	67.8	72.6	75.8	74.2	77.4
	KLD [94]	R50		88.9	83.7	50.1	68.8	78.2	76.1	84.6	89.4	86.2	85.3	63.1	60.9	75.1	71.5	67.5	75.3
		R50	√	88.9	85.2	53.6	81.2	78.2	77.0	84.6	89.5	86.8	86.4	71.7	68.1	76.0	72.2	75.4	78.3
	RetinaNet-O [95]	R50		88.7	77.6	41.8	58.2	74.6	71.6	79.1	90.3	82.2	74.3	54.8	60.6	62.6	69.7	60.6	68.4
		CR50		87.9	74.7	36.5	61.7	77.7	64.9	77.4	90.1	79.6	78.3	54.5	60.4	61.3	57.6	41.3	66.9 (**-1.5**)
		FastTCM-CR50		87.5	77.9	41.0	66.2	70.8	72.5	78.3	89.9	81.6	83.8	55.7	60.3	62.6	71.0	58.1	70.5 (**+2.1**)
	Rotated-FCOS [96]	R50		89.2	72.0	48.0	61.6	79.3	73.5	85.8	90.9	81.1	84.3	59.6	62.7	62.1	69.9	49.3	71.3
		CR50		88.3	73.0	46.0	55.5	78.1	69.0	87.5	90.9	81.5	82.0	57.7	62.6	65.6	59.4	43.6	69.4 (**-1.9**)
		FastTCM-CR50		88.4	77.2	45.8	59.6	81.3	83.1	87.9	90.9	84.9	85.0	57.0	64.8	72.3	77.2	58.3	74.3 (**+3.0**)
	Rotated-ATSS [97]	R50		88.9	79.9	48.7	70.7	75.8	74.0	84.1	90.9	83.2	84.1	60.5	65.1	66.7	70.1	57.8	73.4
		CR50		89.1	75.2	45.9	66.8	78.4	74.7	87.2	90.8	83.1	84.8	53.5	65.1	69.6	64.6	52.8	72.1 (**-1.3**)
		FastTCM-CR50		88.7	80.5	46.7	69.9	81.1	83.5	87.8	90.9	82.7	85.8	60.6	63.6	72.9	78.7	56.2	75.3 (**+1.9**)

R50, R101, and R152 denote ResNet-50, ResNet-101, and ResNet-152, respectively. MS indicates that multi-scale testing is used.
The bold font stand for the performance variance.

TABLE XIII
ABLATION STUDY OF EXPLORATION ON LARGE AMOUNTS OF TRAINING DATA

Method	BB	Training Data	Testing Data	F (%)
DB	R50	Joint data	NightTime-ArT	52.8
DB	CR50	Joint data	NightTime-ArT	58.4
DB	TCM-CR50	Joint data	NightTime-ArT	70.2
	FastTCM-CR50	Joint data	NightTime-ArT	**72.6**

The bold font stand for the best performance.

are provided in appendix, available online. Results are shown in Table XIII. The results show that even with the addition of large amounts of training data, existing methods still show limitations to the nighttime data that is obviously out-of-distribution from the training set. However, integrating FastTCM-CR50 can still perform robustly in such cases, indicating its robust generalization ability.

I. Rotated Object Detection

To further validate the generalization ability of our approach, we adapted it to oriented object detection and evaluated its performance on the widely used DOTA-v1.0 [98] dataset, which is specifically designed for oriented object detection in aerial images. The DOTA-v1.0 dataset consists of 15 common categories, 2806 images, and 188,282 instances. During training, we employed the same configuration as the cooperative methods for rotated object detection. As presented in Table XII, we combined our model with previous approaches for oriented object detection. The results illustrate the consistent improvement by using the proposed FastTCM-CR50 backbone. We guess that the improvement of FastTCM-CR50 originates from its ability to utilize the rich prior knowledge offered by CLIP, thus optimizing the spotting and location of specific categories within satellite images. Specifically, FastTCM-CR50 initiates a synergy between visual features and their textual descriptions. Visual features aligning with textual descriptors are amplified, enabling the visual features to focus more on the segments related to remote sensing categories, thereby augmenting the performance of rotated object detection. Qualitative results on DOTA-v1.0 are presented in Fig. 8.

Fig. 9. Failure cases. Green polygons represent predicted detection results, and blue circle represents error detection regions. The dashed boxes stand for predicted recognition results, and blue characters are error recognition results.

J. Summary of the Experiments

The experimental analysis of FastTCM-CR50 in scene text detection and scene text spotting across various benchmarks demonstrates several advantages: (1) FastTCM can be seamlessly integrated to enhance existing scene text detectors and spotters with high efficiency. (2) FastTCM significantly improves the few-shot training ability of the detectors and spotters. (3) FastTCM also shows powerful generalization ability for generalization tasks, including domain adaptation, NightTime-ArT dataset, and rotated object detection dataset DOTA-v1.0. Some of the failure cases can be visualized in Fig. 9. We can see that some text-like objects might be mistakenly regarded as the positive text region.

V. CONCLUSION

The proposed FastTCM-CR50 backbone provides a notable enhancement to numerous scene text detectors and spotters, achieving consistent performance improvements, along with a significant increase in inference speed of 47.1% compared to previous TCM-CR50. We conduct comprehensive ablation studies to demonstrate the effectiveness every aspect of the proposed method. The robustness of FastTCM-CR50 is also demonstrated by its remarkable few-shot learning capabilities and generalization ability. Significant improvements on the NightTime-ArT subset from ICDAR2019-ArT and the rotated object detection

dataset (DOTA-v1.0) further highlight the potential of the proposed method. We hope this work can provide a foundation for future advancements in the field of scene text detection and spotting.

REFERENCES

[1] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 1–16.

[2] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Conditional prompt learning for vision-language models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16816–16825.

[3] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–20.

[4] Y. Rao et al., "DenseCLIP: Language-guided dense prediction with context-aware prompting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18061–18070.

[5] M. Xu et al., "A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 736–753.

[6] S. Song et al., "Vision-language pre-training for boosting scene text detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15681–15691.

[7] Q. Wan, H. Ji, and L. Shen, "Self-attention based text knowledge mining for text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5979–5988.

[8] C. Xue, W. Zhang, Y. Hao, S. Lu, P. H. S. Torr, and S. Bai, "Language matters: A weakly supervised vision-language pre-training approach for scene text detection and spotting," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 1–19.

[9] W. Yu, Y. Liu, W. Hua, D. Jiang, B. Ren, and X. Bai, "Turning a clip model into a scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023.

[10] B. Shi, X. Bai, and S. J. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3482–3490.

[11] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A flexible representation for detecting text of arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 1–17.

[12] X. Li, W. Wang, W. Hou, R.-Z. Liu, T. Lu, and J. Yang, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9328–9337.

[13] W. Wang et al., "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8439–8448.

[14] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11474–11481.

[15] J. R. Tang et al., "Few could be better than all: Feature sampling and grouping for scene text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4563–4572.

[16] S. Long, S. Qin, D. Panteleev, A. Bissacco, Y. Fujii, and M. Raptis, "Towards end-to-end unified scene text detection and layout analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1039–1049.

[17] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4159–4167.

[18] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3454–3461.

[19] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3066–3074.

[20] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 745–753.

[21] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4161–4167.

[22] X. Zhou et al., "East: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2642–2651.

[23] C. Zhang et al., "Look more than once: An accurate detector for text of arbitrary shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10544–10553.

[24] X. Wang, Y. Jiang, Z. Luo, C.-L. Liu, H. Choi, and S. Kim, "Arbitrary shape scene text detection with adaptive text region representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6442–6451.

[25] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3122–3130.

[26] M. He et al., "MOST: A multi-oriented scene text detector with localization refinement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8809–8818.

[27] S.-X. Zhang, X. Zhu, C. Yang, H. Wang, and X.-C. Yin, "Adaptive boundary proposal network for arbitrary shape text detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 1285–1294.

[28] P. Dai, S. Zhang, H. Zhang, and X. Cao, "Progressive contour regression for arbitrary-shape scene text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7389–7398.

[29] M. Ye, J. Zhang, S. Zhao, J. Liu, B. Du, and D. Tao, "DPText-DETR: Towards better scene text detection with dynamic points in transformer," in *Proc. AAAI Conf. Artif. Intell.*, 2023.

[30] S.-X. Zhang, X. Zhu, C. Yang, and X.-C. Yin, "Arbitrary shape text detection via boundary transformer," *IEEE Trans. Multimedia*, vol. 26, pp. 1747–1760, 2023.

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

[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[33] H. Li, P. Wang, and C. Shen, "Towards end-to-end text spotting in natural scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7266–7281, Oct. 2022.

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

[35] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An end-to-end textspotter with explicit alignment and attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5020–5029.

[36] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5676–5685.

[37] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 67–83.

[38] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 532–548, Feb. 2021.

[39] S. Qin, A. Bissacco, M. Raptis, Y. Fujii, and Y. Xiao, "Towards unconstrained end-to-end text spotting," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4703–4713.

[40] F. Wei, H. Wenhao, Y. Fei, Z. Xu-Yao, and C.-L. Liu, "TextDragon: An end-to-end framework for arbitrary shaped text spotting," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9075–9084.

[41] H. Wang et al., "All you need is boundary: Toward arbitrary-shaped text spotting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12160–12167.

[42] X. Linjie, T. Zhi, H. Weilin, and R. S. Matthew, "Convolutional character networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9125–9135.

[43] M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, "Mask textspotter v3: Segmentation proposal network for robust scene text spotting," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 706–722.

[44] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "ABCNet: Real-time scene text spotting with adaptive bezier-curve network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9806–9815.

[45] S. Fang, Z. Mao, H. Xie, Y. Wang, C. C. Yan, and Y. Zhang, "ABINet++: Autonomous, bidirectional and iterative language modeling for scene text spotting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7123–7141, Jun. 2022.

[46] M. Huang et al., "SwinTextSpotter: Scene text spotting via better synergy between text detection and text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4583–4593.

[47] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[48] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–16.
[49] X. Zhang, Y. Su, S. Tripathi, and Z. Tu, "Text spotting transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9509–9518.
[50] Y. Kittenplon, I. Lavi, S. Fogel, Y. Bar, R. Manmatha, and P. Perona, "Towards weakly-supervised text spotting using a multi-task transformer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4594–4603.
[51] D. Peng et al., "SPTS: Single-point text spotting," in *Proc. ACM Int. Conf. Multimedia*, 2021, pp. 4272–4281.
[52] M. Ye et al., "DeepSolo: Let transformer decoder with explicit points solo for text spotting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 19348–19357.
[53] G. Goh et al., "Multimodal neurons in artificial neural networks," *Distill*, 2021, [Online]. Available: https://distill.pub/2021/multimodal-neurons
[54] F. Petroni et al., "Language models as knowledge bases?," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2019, pp. 1772–1791.
[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[56] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *Int. J. Comput. Vis.*, pp. 2337–2348, 2022.
[57] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
[58] D. Karatzas et al., "ICDAR 2013 robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit.*, 2013, pp. 1484–1493.
[59] D. Karatzas et al., "ICDAR 2015 competition on robust reading," in *Proc. Int. Conf. Document Anal. Recognit.*, 2015, pp. 1156–1160.
[60] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1083–1090.
[61] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognit.*, vol. 90, pp. 337–345, 2019.
[62] C.-K. Ch'ng, C. S. Chan, and C.-L. Liu, "Total-text: Toward orientation robustness in scene text detection," *Int. J. Document Anal. Recognit.*, vol. 23, no. 1, pp. 31–52, 2019.
[63] C.-K. Chng et al., "ICDAR2019 robust reading challenge on arbitrary-shaped text (RRC-ArT)," in *Proc. Int. Conf. Document Anal. Recognit.*, 2019, pp. 1571–1576.
[64] N. Nayef et al., "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification - RRC-MLT," in *Proc. Int. Conf. Document Anal. Recognit.*, 2017, pp. 1454–1459.
[65] N. Nayef et al., "ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition–RRC-MLT-2019," in *Proc. Int. Conf. Document Anal. Recognit.*, 2019, pp. 1454–1459.
[66] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2315–2324.
[67] A. Singh, G. Pang, M. Toh, J. Huang, W. Galuba, and T. Hassner, "TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8798–8808.
[68] S.-X. Zhang, X. Zhu, L. Chen, J.-B. Hou, and X.-C. Yin, "Arbitrary shape text detection via segmentation with probability maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 2736–2750, Mar. 2023.
[69] F. Zhan, C. Xue, and S. Lu, "GA-DAN: Geometry-aware domain adaptation network for scene text detection and recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9104–9114.
[70] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "TextField: Learning a deep direction field for irregular scene text detection," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5566–5579, Nov. 2019.
[71] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9357–9366.
[72] Y. Wang, H. Xie, Z. Zha, M. Xing, Z. Fu, and Y. Zhang, "ContourNet: Taking a further step toward accurate arbitrary-shaped scene text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11750–11759.
[73] S.-X. Zhang et al., "Deep relational reasoning graph network for arbitrary shape text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9696–9705.
[74] Z. Raisi, M. A. Naiel, G. Younes, S. Wardell, and J. S. Zelek, "Transformer-based text detection in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, 2021, pp. 3156–3165.
[75] J. Ye, Z. Chen, J. Liu, and B. Du, "TextFuseNet: Scene text detection with richer fused features," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 516–522.
[76] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 919–931, Jan. 2023.
[77] L. Xing, Z. Tian, W. Huang, and M. R. Scott, "Convolutional character networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9125–9135.
[78] Y. Baek et al., "Character region attention for text spotting," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 504–521.
[79] L. Qiao et al., "Text perceptron: Towards end-to-end arbitrary-shaped text spotting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11899–11907.
[80] W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "TextDragon: An end-to-end framework for arbitrary shaped text spotting," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9075–9084.
[81] W. Wang et al., "PAN++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5349–5367, Sep. 2022.
[82] P. Wang et al., "PGNet: Real-time arbitrarily-shaped text spotting with point gathering network," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2782–2790.
[83] L. Qiao et al., "MANGO: A mask attention guided one-stage scene text spotter," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 2467–2476.
[84] R. Ronen, S. Tsiper, O. Anschel, I. Lavi, A. Markovitz, and R. Manmatha, "GLASS: Global to local attention for scene-text spotting," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 249–266.
[85] Y. Liu et al., "ABCNet v2: Adaptive bezier-curve network for real-time end-to-end text spotting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8048–8064, Nov. 2022.
[86] W. Wu et al., "Synthetic-to-real unsupervised domain adaptation for scene text detection in the wild," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–14.
[87] K.-H. Kim, Y. Cheon, S. Hong, B.-S. Roh, and M. Park, "PVANET: Deep but lightweight neural networks for real-time object detection," 2016, *arXiv:1608.08021*.
[88] M. Liao, B. Song, M. He, S. Long, C. Yao, and X. Bai, "SynthText3D: Synthesizing scene text images from 3D virtual worlds," *Sci. China-Inf. Sci.*, vol. 63, pp. 1–14, 2020.
[89] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.
[90] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, "ST-GAN: Spatial transformer generative adversarial networks for image compositing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9455–9464.
[91] P. Zhao, Z. Qu, Y. Bu, W. Tan, and Q. Guan, "PolarDet: A fast, more precise detector for rotated target in aerial images," *Int. J. Remote Sens.*, vol. 42, no. 15, pp. 5821–5851, 2021.
[92] B. Zhong and K. Ao, "Single-stage rotation-decoupled detector for oriented object," *Remote Sens.*, vol. 12, no. 19, 2020, Art. no. 3262.
[93] X. Yang, J. Yan, Q. Ming, W. Wang, X. Zhang, and Q. Tian, "Rethinking rotated object detection with gaussian wasserstein distance loss," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11830–11841.
[94] X. Yang et al., "Learning high-precision bounding box for rotated object detection via kullback-leibler divergence," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 18381–18394.
[95] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.
[96] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9626–9635.
[97] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9759–9768.
[98] G.-S. Xia et al., "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3974–3983.

Wenwen Yu received the BS and MS degrees from Xuzhou Medical University, China, in 2018 and 2021, respectively. He is currently working toward the PhD degree with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology (HUST), Wuhan, China. He was a research intern with Ping An Property and Casualty Insurance Company of China, and Tencent YouTu Lab, respectively. His current research interests include computer vision, scene text detection, recognition, and understanding.

Yuliang Liu (Member, IEEE) received the BS, and PhD degrees from the South China University of Technology (SCUT), Guangzhou, China, in 2016 and 2020, respectively. He was also the postdoc with the University of Adelaide and the Chinese University of Hong Kong. He is currently a research professor with the School of Artificial Intelligence and Automation, HUST. His research interests include text detection, spotting and analysis.

Xingkui Zhu received the BS degree from Beihang University, in 2018, and the MS degree from the North China Institute of Computing Technology, in 2022. He is currently working toward the PhD degree with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His current research interests lie in computer vision, object detection, and efficient fine-tuning of pre-trained vision models.

Haoyu Cao received the ME degree in computer technology from the Hefei University of Technology. He is currently the senior researcher with Tencent. His current research interest focuses on natural language processing and computer vision.

Xing Sun received the PhD degree from the University of Hong Kong, in 2016. He is currently a principal researcher with Tencent YoutuLab. His research interests include image processing, machine learning, and computer vision.

Xiang Bai (Senior Member, IEEE) received the BS, MS, and PhD degrees in electronics and information engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively. He is a professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology (HUST). He is also the vice-director with the National Center of Anti-Counterfeiting Technology, HUST. His research interests include object recognition, shape analysis, scene text recognition, and intelligent systems. He serves as an associate editor for *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Pattern Recognition*, *Frontiers of Computer Science*, *International Journal on Document Analysis and Recognition*, *China Science: Information Science* and *ACTA AUTOMATICA SINICA*. He is a fellow of IAPR.