Towards Accurate Video Text Spotting with Text-wise Semantic Reasoning

Xinyan Zu, Haiyang Yu, Bin Li*, Xiangyang Xue
Shanghai Key Laboratory of Intelligent Information Processing
School of Computer Science, Fudan University
{xyzu20, hyyu20, libin, xyxue}@fudan.edu.cn

Abstract

Video text spotting (VTS) aims at extracting texts from videos, where text detection, tracking and recognition are conducted simultaneously. There have been some works that can tackle VTS; however, they may ignore the underlying semantic relationships among texts within a frame. We observe that the texts within a frame usually share similar semantics, which suggests that, if one text is predicted incorrectly by a text recognizer, it still has a chance to be corrected via semantic reasoning. In this paper, we propose an accurate video text spotter, VLSpotter, that reads texts visually, linguistically, and semantically. For 'visually', we propose a plug-and-play text-focused super-resolution module to alleviate motion blur and enhance video quality. For 'linguistically', a language model is employed to capture intra-text context to mitigate wrongly spelled text predictions. For 'semantically', we propose a text-wise semantic reasoning module to model inter-text semantic relationships and reason for better results. The experimental results on multiple VTS benchmarks demonstrate that the proposed VLSpotter outperforms the existing state-of-the-art methods in end-to-end video text spotting.

1 Introduction

In recent years, we have witnessed rapid growth in new methods [Wang *et al.*, 2017; Cheng *et al.*, 2019; Cheng *et al.*, 2020; Wu *et al.*, 2021; Wu *et al.*, 2022] and datasets [Karatzas *et al.*, 2013; Karatzas *et al.*, 2015; Wu *et al.*, 2021; Cheng *et al.*, 2019] for video text spotting (VTS). It has been drawing increasing research interest due to its crucial role in computer vision tasks, such as video retrieval [Spolaor *et al.*, 2020] and video subtitle recognition [Xu *et al.*, 2018], and downstream industrial applications, such as license plate recognition [Shashirangana *et al.*, 2020] and machine translation [Stahlberg, 2020].

There have been some works [Wang *et al.*, 2017; Cheng *et al.*, 2019; Cheng *et al.*, 2020; Wu *et al.*, 2021; Wu *et al.*, 2022] aiming to tackle the VTS problem, most of which tend to design a multi-task pipeline containing text detection, tracking and recognition as three major components. For example, a Transformer-based VTS model is proposed to boost performance in [Wu *et al.*, 2021]. Some methods [Cheng *et al.*, 2019; Cheng *et al.*, 2020] utilize a scoring mechanism to save recognition time. In addition, a light-weighted VTS model is proposed in [Wu *et al.*, 2022] to achieve real-time inference. However, these methods mainly focus on improving the backbones of models and may ignore the underlying semantic relationships among texts within a frame. In this case, most existing methods leverage only two types of information, *i.e.*, visual information and temporal information.

Figure 1: Examples of inter-text semantic relationships in a frame.

In addition to the aforementioned two types of helpful information, inter-text semantic relationships among texts within a frame may also be beneficial for improving the performance of VTS. For example, the first frame in Fig. 1 is captured from a food store, where the texts therein are related to the semantic concept of food. One can consider a situation that 'Steak' and 'Panini' are correctly predicted, while 'Hamburger' is wrongly predicted as 'lamberger' due to occlusion. As the texts within this particular frame share similar semantics, humans can naturally figure out that 'lamberger' is wrongly spelled based on the observation that both 'Steak' and 'Panini' belong to the semantic concept of food, and can further infer the correct text 'Hamburger' belonging to the same semantic concept. There exist a non-ignorable number of such cases in scene text videos, which hatches our core motivation: We can introduce inter-text semantic reasoning into VTS to achieve more accurate results.

Based on this motivation, we propose an accurate video text spotter, VLSpotter, to read texts visually, linguistically, and semantically. Following the fashion of existing VTS

*Corresponding author

methods [Wang *et al.*, 2017; Cheng *et al.*, 2019; Cheng *et al.*, 2020; Wu *et al.*, 2021; Wu *et al.*, 2022], VLSpotter adopts the paradigm of performing text detection, tracking and recognition simultaneously. In addition to leveraging visual and temporal information carried in text videos, the unique design of the proposed VLSpotter further explores the potential of inter-text semantic relationships. Specifically, for 'visually', we propose a text-focused super-resolution (TFSR) module to effectively alleviate motion blur and produce high-quality frames for subsequent detection, tracking and recognition; moreover, we propose a mask-adaptive super-resolution loss to supervise TFSR, making it focus more on text areas. For 'linguistically', a language model is employed to mitigate wrongly predicted texts through capturing intra-text context. For 'semantically', we propose a text-wise semantic reasoning (TWSR) module to model the semantic relationships among texts with similar semantics. To train the proposed TWSR, we categorize 21K commonly used texts into 20 semantic concepts, and perform semantic reasoning among texts that belong to the same concept. In this way, better recognition results can be obtained with fused visual, linguistic and semantic features.

Extensive experiments of text spotting, tracking and detection are conducted on three VTS benchmarks, ICDAR2013 Video [Karatzas *et al.*, 2013], ICDAR2015 Video [Karatzas *et al.*, 2015], and BOVText [Wu *et al.*, 2021], to evaluate the effectiveness of the proposed VLSpotter. The experimental results demonstrate that VLSpotter outperforms the existing methods by a clear margin under the criterion of end-to-end text spotting, and achieves comparable performance under the criterion of text detection and tracking. The code of VLSpotter is available at GitHub[1].

The contributions of this paper are three-fold:

- Based on the observation that texts within a video frame usually exhibit similar semantics, we propose a text-wise semantic reasoning module to model inter-text semantic relationships.
- We propose a plug-and-play text-focused super-resolution module to enhance video quality and design a mask-adaptive super-resolution loss to force this module to focus more on text areas.
- The proposed VLSpotter outperforms the state-of-the-art methods by a clear margin on ICDAR2015 Video and BOVText in end-to-end text spotting.

2 Related Works

In this section, we first introduce text recognition. Then, we introduce the recent development of end-to-end video text spotting. Finally, we briefly introduce the text image super-resolution task.

2.1 Text Recognition

Some existing text recognition works have introduced semantic reasoning to text recognition. [Yu *et al.*, 2020] proposes a Transformer-based global semantic reasoning module to capture global semantic context via parallel transmissions; its semantic context is actually the embedded features of attended visual features aligned to characters. Additionally, [Bhunia *et al.*, 2021] develops a multi-stage and multi-scale attention decoder to perform reasoning procedure. Different from [Yu *et al.*, 2020], this method conducts a joint visual and semantic reasoning. Inspired by [Bhunia *et al.*, 2021], [He *et al.*, 2022] proposes a reasoning module implemented based on graph convolution to infer the text results with both visual and textual (language) features in an iterative manner. Technically, the aforementioned works conduct semantic reasoning on top of intra-text semantics, thus still ignoring inter-text semantic relationships. Speaking of the diversity of languages in text recognition, a large portion of Chinese texts appear in the VTS benchmarks such as BOVText [Wu *et al.*, 2021] and LSVTD [Cheng *et al.*, 2019]. Some recently proposed Chinese text recognition methods [Yu *et al.*, 2022; Zu *et al.*, 2022; Yu *et al.*, 2021a; Su *et al.*, 2023; Chen *et al.*, 2021b] are capable of tackling them.

2.2 End-to-End Video Text Spotting

Spotting text from videos has already been studied before the era of deep learning. Recently, the rapid growth of deep learning techniques and datasets for text detection and recognition have hatched several new approaches to end-to-end video text spotting. In [Wang *et al.*, 2017], temporal information is used to enable multi-frame tracking, thus improving spotting performance. Subsequently, [Cheng *et al.*, 2019] and [Cheng *et al.*, 2020] design a scoring network to select high-quality instances from the tracking stream, thus allowing their methods to recognize only once. The framework for tracking, recognition, and scoring is also made end-to-end in these works. Recently, [Wu *et al.*, 2021] employs a Transformer-based detection and tracking system to spot arbitrarily oriented texts in videos, and brings the end-to-end spotting performance to a higher level. [Wu *et al.*, 2022] develops an extremely efficient solution while not harming the performance by conducting GPU-parallel detection post-processing. [Wu *et al.*, 2022] introduces contrastive learning to VTS for managing long-range dependencies across multiple frames. Different from these methods, the proposed VLSpotter attempts to leverage the inter-text semantic relationships within video frames.

2.3 Text Image Super-Resolution

Image super-resolution aims to enhance the quality of low-resolution images, and it is also effective in blurred situations. However, generic image super-resolution methods like [Chen *et al.*, 2022b] have limited effect on text regions. To fill this gap, [Mou *et al.*, 2020] employs super-resolution blocks to benefit text recognizer. [Chen *et al.*, 2021a] makes use of text position and content priors; and [Ma *et al.*, 2023] further leverage text probability priors in a concurrent manner. [Chen *et al.*, 2022a] reconstructs high-quality text images by focusing on text strokes. However, these methods only work on text-line images instead of an entire scene image with multiple texts, which is highly demanded in VTS tasks. In this paper, the proposed text-focused super-resolution module aims to handle text-focused super-resolution of the entire image.

[1]https://github.com/FudanVI/FudanOCR/VLSpotter

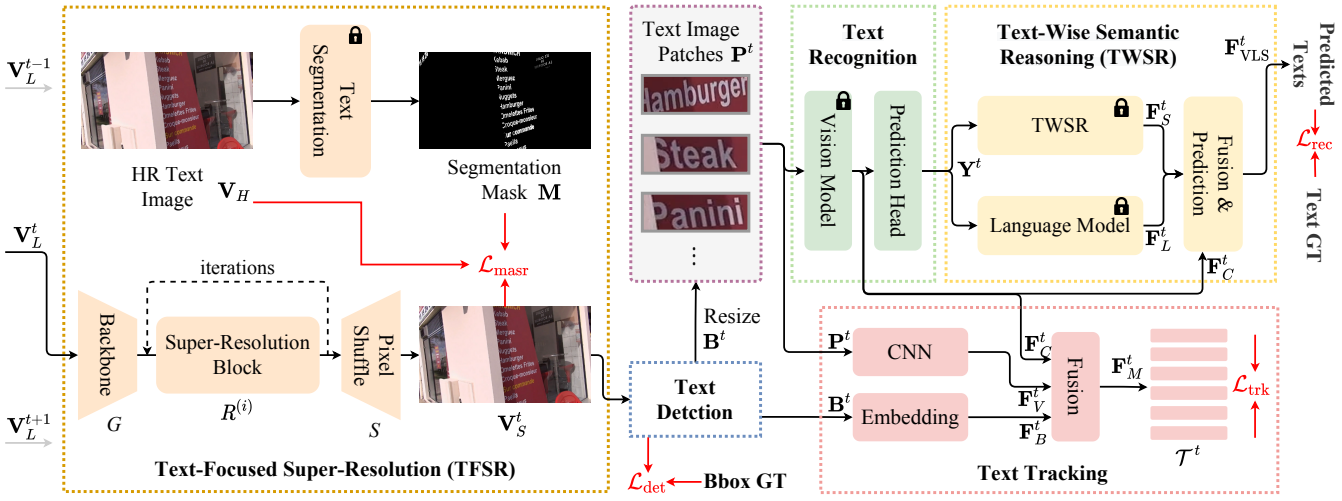


Figure 2: The overall architecture of the proposed VLSpotter. Data flow only used for training is marked in red. The modules that require pre-training are marked with the symbols of a black lock.

3 Methodology

We introduce the details of the proposed VLSpotter in the following. First, we present the proposed Text-Focused Super-Resolution (TFSR) module, which is developed to alleviate the motion blur in text videos and enhance video quality. We next introduce the procedures for text detection, recognition and tracking in VLSpotter. Finally, we introduce the proposed Text-Wise Semantic Reasoning (TWSR) module for correcting text predictions with inter-text semantic relationships within a frame. The overall architecture of VLSpotter is shown in Fig. 2.

3.1 Text-Focused Super-Resolution

To avoid the side effect of motion blur for text detection and recognition, the previous methods [Cheng *et al.*, 2019; Cheng *et al.*, 2020] design a scoring mechanism to select a frame with the best quality in the temporal tracking stream to perform one-time text recognition. However, when all frames in a text video are blurred and achieve low scores, the quality of text areas in the selected frame may still pose challenges for subsequent text detection and recognition. Therefore, we introduce a plug-and-play Text-Focused Super-Resolution (TFSR) module to conquer motion blur in text videos, thus improving the quality of input for subsequent modules.

Considering that super-resolving frames of text video only plays as a preprocessing role for the subsequent modules, we adopt a light-weighted super-resolution framework [Chen *et al.*, 2022b] as the main body of TFSR to reduce the time overhead. Specifically, TFSR consists of a ResNet-based backbone G, a super-resolution block R, and a pixel-shuffle module S. The entire process of TFSR can be formulated as:

$$\mathbf{V}_S^t = S(R^{(i)}(G(\mathbf{V}_L^t))) \tag{1}$$

where $R^{(i)}$ represents that the image features extracted by $G(\cdot)$ are iteratively processed by the super-resolution block R for multiple iterations. Eventually, the input frames $\mathbf{V}_L^t$ with the size of $H \times W$ are super-resolved into the corresponding super-resolution frame $\mathbf{V}_S^t$ with the size of $2H \times 2W$.

The TFSR module aims to obtain the frames with better quality for subsequent text detection and recognition. However, existing datasets for super-resolution mainly focus on scene images with few texts. Although TextZoom [Wang *et al.*, 2020] is proposed to solve the text image super-resolution task, the samples are all cropped text areas rather than scene images with multiple texts. Therefore, it is necessary to build a super-resolution dataset for text images to train the proposed TFSR module. As it lacks a scene-level dataset for TSFR training, we synthesize a dataset by replacing the text areas in video frames with the high-resolution (HR) and low-resolution (LR) images in TextZoom to construct a HR-LR pair of text images. In addition, we also observe that the image-quality labels of frames are available in ICDAR2013 [Karatzas *et al.*, 2013], ICDAR2015 [Karatzas *et al.*, 2015], and LSVTD [Cheng *et al.*, 2019]. Thus, two adjacent frames with different quality scores in these datasets can be regarded as a HR-LR text image pair. Through the aforementioned two strategies, we can construct a sufficient super-resolution dataset, containing both synthetic and real-world samples, for text images to train the proposed TFSR.

To make TFSR focus on text areas, we further propose a mask-adaptive super-resolution (MASR) loss $\mathcal{L}_{\text{masr}}$ to supervise this module. Specifically, we first perform text segmentation on an HR text image to obtain the corresponding text mask $\mathbf{M}$, where a pre-trained text segmentation model [Xu *et al.*, 2021] is employed. Then, the text mask $\mathbf{M}$ is utilized to distinguish text and non-text areas in the text image and further adaptively adjust the loss weight of them. The MASR loss can be computed by:

$$\begin{aligned}\mathcal{L}_{\text{masr}} \;=\; & \tanh(\text{MSE}(\mathbf{V}_H, \mathbf{V}_S)) \cdot \\ & \text{MSE}(\mathbf{V}_H \odot (1-\mathbf{M}), \mathbf{V}_S \odot (1-\mathbf{M})) \\ & +\epsilon \cdot \text{MSE}(\mathbf{V}_H \odot \mathbf{M}, \mathbf{V}_S \odot \mathbf{M})\end{aligned} \tag{2}$$

where $\mathbf{V}_H$ and $\mathbf{V}_S$ represent the HR and SR text images, re-

spectively, $\odot$ denotes the pixel-wise product, and ϵ is a hyper-parameter. Based on the proposed MASR loss, TFSR module could pay more attention to text areas as the super-resolution quality of the whole text image improves.

3.2 Text Detection, Recognition and Tracking

The output of the TFSR module, *i.e.* high-quality video frames, are then fed into the pipeline of subsequent text detection, tracking and recognition.

Detection. Taking the super-resolved frame $\mathbf{V}_S^t$ as input, the detection head [Wang *et al.*, 2021] Det$(\cdot)$ is utilized to predict the location of each text area:

$$\mathbf{B}^t = \text{Det}(\mathbf{V}_S^t) \tag{3}$$

where $\mathbf{B}^t = \{b_1^t, b_2^t, ..., b_N^t\}$ denotes all the bounding boxes predicted by detector Det$(\cdot)$ in the t-th frame; b_n^t and N denote the n-th bounding box and the total number of detected text boxes, respectively.

Recognition. According to $\mathbf{B}^t$, we can obtain N RoIs cropped from $\mathbf{V}_S$. Through transformation, these RoIs can be resized into fix-sized text image patches, denoted as $\mathbf{P}^t = \{p_1^t, p_2^t, ..., p_N^t\}$, which are further sent to the text recognition model. For text recognition, we employ the vision model proposed in [Fang *et al.*, 2021], which consists of a visual feature extractor VM$(\cdot)$ and a prediction head. VM$(\cdot)$ is used to extract the visual features of text image patches $\mathbf{F}_C^t$:

$$\mathbf{F}_C^t = \text{VM}(\mathbf{P}^t) \tag{4}$$

Then, the extracted features $\mathbf{F}_C^t$ go through the prediction head to obtain the coarse text prediction $\mathbf{Y}^t = \{y_1^t, y_2^t, ..., y_N^t\}$:

$$\mathbf{Y}^t = \text{Softmax}(\mathbf{W}\mathbf{F}_C^t + b) \tag{5}$$

Subsequently, the coarse text prediction $\mathbf{Y}^t$ will be sent to a pre-trained language model and the proposed TWSR module in a parallel manner. The language model conducts intra-text spell correction and generates intra-text semantic features $\mathbf{F}_L^t$; and the TWSR module will be detailed in Section 3.3.

Tracking. As shown in Fig. 2, three types of features are used for text tracking: the visual features $\mathbf{F}_V^t$, the contextual features $\mathbf{F}_C^t$ from the vision model VM$(\cdot)$, and the embedding of the coordinates of texts $\mathbf{F}_B^t$. The visual features $\mathbf{F}_V^t$ are extracted by a CNN-based backbone for text tracking. For the purpose of introducing more text-related features to improve the performance of tracking, we fuse the extracted contextual features $\mathbf{F}_C^t$ from VM$(\cdot)$. In addition, the position information of a text is also crucial for text tracking since the positional offset of the text in adjacent frames is tiny. Therefore, we feed the predicted coordinates of texts $\mathbf{B}^t$ in the text detector to an embedding layer to obtain the corresponding features $\mathbf{F}_B^t$. Finally, we concatenate the three types of features and employ a 1×1 convolution layer to fuse them, which can be represented as follows:

$$\mathbf{F}_M^t = \text{Conv}_{1\times1}(\text{Concat}[\mathbf{F}_B^t : \mathbf{F}_V^t : \mathbf{F}_C^t]) \tag{6}$$

where $\mathbf{F}_M^t$ denotes the fused features for tracking. In the inference stage, we employ the KM [Kuhn, 1955] algorithm to match the texts between two adjacent frames, and thus creating a tracking trajectory for each text:

$$\mathcal{T}^t = \text{KM}(\mathbf{F}_M^t, \mathbf{F}_M^{t-1}) \tag{7}$$

where $\mathcal{T}^t$ denotes the set of all matched texts from the $t-1$th frame and the detected new texts in the current frame.

To further increase the effectiveness of $\mathbf{F}_M^t$, we train the learnable fusion parameters in Eq. 6 by maximizing the similarity of the fused tracking features $\mathbf{F}_M$ between temporal-adjacent texts (positive pairs) that belong to the same tracking trajectory, while minimizing the similarity of the fused tracking features $\mathbf{F}_M$ between text pairs that do not exist in the same trajectory (negative pairs). The contrastive loss is capable of unifying both the positive and the negative objectives above: Let $\mathcal{T}^t = \{\tau_1, ..., \tau_L\}$ where τ_l denotes the l-th text, the tracking loss of the trajectory can be computed by:

$$\mathcal{L}_{\text{trk}} = -\frac{1}{L}\sum_{\tau_l \in \mathcal{T}^t} \log \frac{e^{\text{sim}(\mathbf{F}_M^t(\tau_l), \mathbf{F}_M^t(\tau_{l-1}))/\sigma}}{\sum_{\tau_k \notin \mathcal{T}^t} e^{\text{sim}(\mathbf{F}_M^t(\tau_l), \mathbf{F}_M^t(\tau_k))/\sigma}} \tag{8}$$

where $\tau_k \notin \mathcal{T}^t$ denotes a text in the other trajectories and σ is a temperature parameter. In the beginning of the given trajectory ($t = 1$) , the positive pair is set to be τ_l and itself.

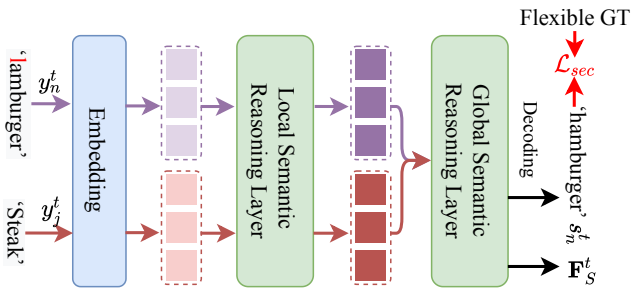


Figure 3: Details of the proposed TWSR module. This module mainly consists of an embedding layer, a local semantic reasoning layer, and a global semantic reasoning layer.

3.3 Text-Wise Semantic Reasoning

The prediction results outputted by the text recognizer may be wrongly spelled due to occlusion or other reasons. However, based on our observation that texts within a video frame usually exhibit similar semantics, which suggests that the wrongly predicted texts may be corrected via semantic reasoning. To this end, we propose a text-wise semantic reasoning (TWSR) module to model inter-text semantic relationships among the texts within a frame.

Let y_n^t denote the n-th text prediction in $\mathbf{Y}^t$, and y_j^t denote an arbitrary text in $\mathbf{Y}^t$ other than y_n^t, *i.e.*, $1 \le j \ne n \le N$. As shown in Fig. 3, the TWSR module takes a pair of texts (y_n^t, y_j^t) as input, where the support text y_j^t may have semantic relationships with the coarsely predicted text y_n^t. Subsequently, TWSR processes the input text pair through an embedding layer, a local semantic reasoning layer, and a global semantic reasoning layer. After going through these layers, the inter-text semantic features $\mathbf{F}_S^t$ between y_n^t and y_j^t can be extracted. Finally, TWSR comes up with a single text output

s_n^t. We expect the semantic reasoning result s_n^t to be the correction of y_n^t, if y_n^t is wrongly predicted, or to remain y_n^t, if y_n^t and y_j^t are irrelevant in semantics.

To achieve this goal, we construct a text-to-concept (T2C) taxonomy, which can be used for training TWSR to successfully model the inter-text semantic relationships. The data of T2C are collected from WordNet, Wikipedia, and Amazon Reviews[2], including 20 concepts (food, feeling, animal, location, art, movie, traffic, weather, act, event, brand, sport, celebrity, book, clothing, health, science, politic, travel and date), which comprises more than 21K words. Each word in T2C belongs to a specific concept (e.g. 'apple–food' and 'dog–animal').

Based on the collected T2C taxonomy, we propose a tailored pre-training strategy to provide flexible supervision for TWSR. Let c_q denote the q-th word in the concept set $\mathcal{C}^{\text{cpt}}$, which contains all words with the same semantic concept 'cpt'. The insight of our strategy is to set a flexible ground truth $\hat{x}$ that alters conditionally as follows:

$$\hat{x} = \begin{cases} x, & \min_q(\mathcal{E}(x, c_q)) \geq \phi \\ c_{\text{argmin}_q(\mathcal{E}(x,c_q))}, & \text{otherwise} \end{cases} \tag{9}$$

where $\mathcal{E}(\cdot)$ denotes the character-wise edit-distance and ϕ is a threshold. In the training stage, x is randomly selected from the T2C taxonomy during training and the semantic support text of x, namely c_q, should be randomly selected from $\mathcal{C}^{\text{cpt}}$. Random noises will be manually added to the input text x to mimic the wrongly spelled situations, where the edit-distance between the noised text and the original text is constrained to be smaller than the threshold ϕ. In this case, if the shortest edit-distance $\min_q(\mathcal{E}(x, c_q))$ is smaller than ϕ, $c_{\text{argmin}_q(\mathcal{E}(x,c_q))}$ is taken as the ground truth. Otherwise, if $\min_q(\mathcal{E}(x, c_q))$ is too large, indicating that x does not belong to the concept 'cpt', we use x itself as the ground truth to encourage the TWSR to turn off the reasoning mode and maintain the original coarse predicted texts from the vision model. During the pre-training stage, the cross-entropy loss $\mathcal{L}_{sec}$ is adopted to supervise this module.

In inference, the target text y_n^t and each support text y_j^t are input into the TWSR module to obtain the inter-text semantic features $\mathbf{F}_S^t$. Following the fashion of ABInet [Fang *et al.*, 2021], a gated fusion operation is adopted to fuse three types of features: visual contextual features $\mathbf{F}_C^t$, intra-text semantic features $\mathbf{F}_L^t$, and inter-text semantic features $\mathbf{F}_S^t$. Finally, the fused visual, linguistic and semantic features $\mathbf{F}_{\text{VLS}}^t$ can be produced and further used for final prediction, which is supervised by the cross-entropy loss $\mathcal{L}_{\text{rec}}$.

3.4 Overall Objective

Except for the aforementioned pre-training losses $\mathcal{L}_{\text{masr}}$ for TFSR and $\mathcal{L}_{\text{sec}}$ for TWSR, the rest modules of VLSpotter are jointly trained as follows:

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{trk}} + \lambda_1 \mathcal{L}_{\text{det}} + \lambda_2 \mathcal{L}_{\text{rec}} \tag{10}$$

where λ_1 and λ_2 are hyper-parameters for trade-off.

[2]https://s3.amazonaws.com/amazon-reviews-pds/readme.html

Methods	End-to-End Video Text Spotting (%)					
	ID_{F_1}	MOTA	MOTP	M-M	M-L	FPS
ICDAR2015 Video						
USTB-TV	21.3	13.2	66.6	6.6	67.7	-
StradVision-1	28.2	15.6	68.5	9.5	60.7	-
USTB-TV(2)	32.0	9.0	70.2	8.9	59.5	-
FREE	61.9	53.0	74.9	45.5	35.9	8.8
TransVTSpotter	61.5	53.2	74.9	-	-	9.0
CoText	72.0	59.0	74.5	48.6	26.4	**41.0**
VLSpotter	**72.8**	**60.1**	**76.4**	**50.0**	**26.1**	15.9
BOVText						
EAST+CRNN	6.8	-79.3	76.3	-	-	-
PSENet+CRNN	31.3	-17.0	79.2	-	-	-
DB+CRNN	38.3	-13.2	81.3	-	-	-
TransVTSpotter	43.6	-1.4	82.0	-	-	9.0
CoText	48.3	**11.4**	80.3	32.8	62.1	**36.2**
VLSpotter	**48.8**	10.5	**82.9**	**34.9**	**59.9**	14.0

Table 1: Performance comparison with state-of-the-art methods for end-to-end text spotting. The proposed VLSpotter outperforms the compared methods in most metrics.

4 Experiments

In this section, we first introduce the adopted VTS datasets and the implementation details of the proposed VLSpotter. Subsequently, we compare VLSpotter with the state-of-the-art methods on each VTS benchmark for end-to-end video text spotting, video text tracking and detection. Finally, ablation studies are conducted to evaluate the effectiveness of the proposed modules in VLSpotter.

4.1 Datasets

We conduct experiments on three commonly-used datasets: ICDAR2013 Video [Karatzas *et al.*, 2013], ICDAR2015 Video [Karatzas *et al.*, 2015], and BOVText [Wu *et al.*, 2021].

- **ICDAR2013 Video** was published in the ICDAR Robust Reading Competition. It contains 28 wild videos captured by a moving camera, where 13 videos are used for training and 15 videos for testing. The total number of frames is 15,277, containing 93,934 texts. This dataset only contains English texts.
- **ICDAR2015 Video** is the upgrade revision of the ICDAR2013 Video dataset, and the number of videos expands to 49, where 25 videos for training and 24 videos for testing. The total number of frames is 27,824, containing 143,588 texts. Both ICDAR2013 and ICDAR2015 Video datasets are quite challenging due to frequent motion blurs.
- **BOVText** is the largest video text spotting dataset that includes 2,021 text videos in a variety of scenarios. This dataset covers various scenarios, e.g., life vlog, driving, movie, etc. It contains 1,757,598 frames with 7,292,261 texts, where 1,328,575 frames from 1,541 videos are used for training and 429,023 frames from 480 videos are used for testing.

4.2 Implementation Details

The proposed VLSpotter is implemented with PyTorch. All experiments are conducted on a single RTX 3090 GPU with

Methods	Video Text Tracking (%)					
	ID_{F_1}	MOTA	MOTP	M-M	M-L	FPS
ICDAR2013 Video						
YORO	62.5	47.3	73.7	33.1	45.3	14.3
SVRep	65.1	53.2	**76.7**	38.2	33.2	17.8
CoText	**68.1**	**55.8**	76.4	44.6	28.7	**41.3**
VLSpotter	67.8	52.9	75.0	**45.8**	**26.4**	16.0
ICDAR2015 Video						
USTB-TV	25.9	7.4	70.8	7.4	66.1	-
StradVision-1	25.9	7.9	70.2	6.5	70.8	-
USTB-TV(2)	21.9	12.3	71.8	4.8	72.3	-
AJOU	36.1	16.4	72.7	14.1	62.0	-
FREE	57.9	43.2	**76.7**	36.6	44.4	8.8
TransVTSpotter	57.3	44.1	75.8	34.3	33.7	9.0
SVRep	66.1	49.5	73.9	44.9	27.1	13.4
CoText	**68.6**	**51.4**	73.6	**49.6**	**23.5**	**41.0**
VLSpotter	67.5	50.8	74.0	45.2	24.7	15.9
BOVText						
EAST	28.1	-21.6	75.8	-	-	-
PSENet	45.9	52.1	77.5	-	-	-
DB	48.3	53.2	78.3	-	-	-
TransVTSpotter	64.7	68.2	82.1	57.3	31.4	9.0
SVRep	75.4	69.3	**84.5**	59.0	29.7	12.2
CoText	77.3	**70.0**	81.1	**61.1**	**23.7**	**36.2**
VLSpotter	**77.4**	69.8	80.5	59.9	24.7	14.0

Table 2: Performance comparison with state-of-the-art methods for text tracking. Compared with these methods, the proposed VLSpotter achieves comparable performance on the three datasets.

24GB memory. For VLSpotter, the TFSR module, the TWSR module, and the vision model are all pre-trained in advance for better results. The language model used in VLSpotter is off-the-shelf and frozen during training. We use the Adadelta optimizer with an initial learning rate 0.1, which further shrinks every 200 epochs. Based on the empirical experiments, we set the hyper-parameters ε, σ, ϕ, λ_1, λ_2 to 1, 0.5, 3, 1, 1, respectively.

4.3 Performance Comparison

Following [Wu *et al.*, 2021], we adopt ID_{F_1}, MOTA, MOTP, M-Matched (M-M), and M-Lost (M-L) as the evaluation metrics to evaluate the performance of VLSpotter and the compared methods in video text spotting and tracking. Precision, Recall, and F-Measure are used to evaluate the performance in video text detection. In addition, FPS is also reported to evaluate the efficiency of the compared methods.

End-to-End Video Text Spotting. In VTS, improving the performance of end-to-end video text spotting is the most crucial and ultimate goal. Following the fashion of existing works, we compare VLSpotter with six existing methods (*i.e.*, USTB-TV/USTB-TV(2) [Karatzas *et al.*, 2015], StardVision-1 [Karatzas *et al.*, 2015], FREE [Cheng *et al.*, 2020], TransVTSpotter [Wu *et al.*, 2021] and CoText [Wu *et al.*, 2022]) on the ICDAR2015 Video dataset. The detailed experimental results are reported in Tab. 1. The proposed VLSpotter achieves the best spotting performance in most metrics. Compared to the most recent state-of-the-art method CoText, the results of VLSpotter in ID_{F_1}, MOTA, MOTP, M-Matched, and M-Lost are boosted by 0.8%, 1.1%, 1.9%, 1.4%, and 0.3%, respectively. On the BOVText dataset, except for the result in MOTA, the proposed VLSpotter still outperforms the state-of-the-art method CoText by 0.5%, 2.6%, 2.1%, and 2.2% in ID_{F_1}, MOTP, M-Matched, and M-Lost, respectively.

Methods	Video Text Detection (%)			
	Precision	Recall	F-Measure	FPS
Epshtein *et.al.*	39.8	32.5	35.9	-
Zhao *et.al.*	46.3	47.0	46.7	-
Yin *et.al.*	48.6	54.7	51.6	-
Khare *et.al.*	57.9	55.9	51.7	-
Wang *et.al.*	58.3	51.7	54.5	-
Shivakumar *et.al.*	61.0	57.0	59.0	-
Wu *et.al.*	63.0	68.0	65.0	-
Yu *et.al.*	82.4	56.4	66.9	-
ASGD	75.5	64.1	69.3	9.6
FREE	79.7	68.4	73.6	8.8
SVRep	81.2	68.3	74.2	13.5
CoText	**82.6**	71.6	**76.7**	**41.3**
VLSpotter	82.3	**71.8**	76.0	16.0

Table 3: Performance comparison with state-of-the-art methods for video text detection on ICDAR2013 Video. In the video text detection setting, the proposed VLSpotter can achieve the best performance in Recall.

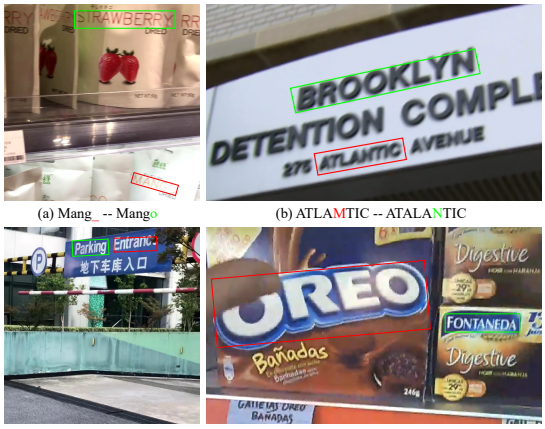
To validate the effectiveness of inter-text semantic reasoning, we demonstrate four examples that the TWSR module successfully corrects wrongly predicted texts in Fig. 4. The presented examples show different scenarios related to food, traffic, location, and brand, respectively. Through the visualization, we observe that the texts in a frame indeed exist semantic relationships among one another, which is fully explored in our method to improve the performance of text spotting. For example, 'UREO' is corrected to 'OREO' with the help of 'FONTANEDA' (shown in Fig. 4(d)).

Video Text Tracking. As shown in Tab. 2, we evaluate the tracking performance of VLSpotter on ICDAR2013 Video, ICDAR2015 Video, and BOVText, and adopt multiple state-of-the-art text tracking methods, *i.e.*, USTB-TV/USTB-TV(2) [Karatzas *et al.*, 2015], StradVision-1 [Karatzas *et al.*, 2015], AJOU [Koo and Kim, 2013], FREE [Cheng *et al.*, 2020], TransVTSpotter [Wu *et al.*, 2021], SVRep [Li *et al.*, 2021], CoText [Wu *et al.*, 2022], YORO [Cheng *et al.*, 2019], EAST [Zhou *et al.*, 2017], PSENet [Wang *et al.*, 2019], and DB [Liao *et al.*, 2020] for comparison. According to the experimental results, VLSpotter achieves comparable tracking performance with the state-of-the-art method CoText; while on challenging BOVText dataset, VLSpotter even outperforms CoText by 0.1% in ID_{F_1}. Although CoText models the long-range temporal dependency with the tailored contrastive loss, VLSpotter can still achieve comparable performance in video text tracking, which may benefit from the high-quality frames produced by the TFSR module.

Video Text Detection. In video text detection, we evaluate the performance of VLSpotter on ICDAR2013 Video. Multiple state-of-the-art text detectors, *i.e.*, [Epshtein *et al.*, 2010], [Zhao *et al.*, 2010], [Yin *et al.*, 2013], [Khare *et al.*, 2017], [Wang *et al.*, 2018], [Shivakumara *et al.*, 2017], [Wu *et al.*, 2015], [Yu *et al.*, 2021b], ASGD [Feng *et al.*, 2021],

Model	Video Text Spotting (%)			Video Text Tracking (%)			Video Text Detection (%)		
	ID_{F_1}	MOTA	MOTP	ID_{F_1}	MOTA	MOTP	Precision	Recall	F-Measure
Baseline	70.2	58.8	73.1	65.9	48.2	72.5	78.9	69.3	74.0
Baseline + TFSR	70.6	59.1	73.8	67.2	50.6	73.8	82.0	71.7	75.8
Baseline + TWSR	72.4	59.7	76.2	66.0	48.1	72.5	78.9	69.4	74.2
Baseline + TWSR + TFSR	72.8	60.1	76.4	67.5	50.8	74.0	82.3	71.8	76.0

Table 4: Ablation study of VLSpotter. The experiments of video text spotting and tracking are conducted on ICDAR2015 Video, and the experiments of video text detection are conducted on ICDAR2013 Video.



(a) Mang_ -- Mango (b) ATLAMTIC -- ATALANTIC

(c) Entranc_ -- Entrance (d) UREO -- OREO

Figure 4: Visualization of the effectiveness of inter-text semantic reasoning in frames that contain different semantic concepts: (a) Food, (b) Location, (c) Traffic, and (d) Brand. Red bounding boxes denote wrongly predicted texts, which are finally corrected through semantic reasoning by the support texts in green bounding boxes.

FREE [Cheng *et al.*, 2020], SVRep [Li *et al.*, 2021] and CoText [Wu *et al.*, 2022] are used for comparison, and the experimental results are shown in Tab. 3. According to the experimental results, the proposed VLSpotter is tied with the state-of-the-art method CoText [Wu *et al.*, 2022]. Specifically, VLSpotter is superior in Recall, but it slightly falls behind CoText in Precision and F-Measure.

4.4 Ablation Study

The proposed VLSpotter contains a pluggable super-resolution module TFSR and a semantic reasoning module TWSR. Therefore, it is important to investigate the effectiveness of both modules. As shown in Tab. 4, we conduct ablation experiments to measure the contributions of TFSR and TWSR. Unsurprisingly, if both of them are unequipped, the performance of VLSpotter will decrease by nearly 2% in all metrics. On one hand, when TWSR is equipped to the baseline model, the performance on recognition-related tasks can be improved clearly. On the other hand, the performance can be boosted in a comprehensive manner by equipping TFSR because it generally increases the visual quality of the input frame and suppresses motion blurs.

4.5 Temporal Semantic Relationships

Intuitively, it is likely that there also exist inter-text semantic relationships between texts from temporal-adjacent frames, considering that scenes of adjacent frames are usually the same ones. Tab. 5 shows interesting results of establishing inter-text semantic reasoning among temporal-adjacent frames. In this case, by sacrificing inference speed, the performance of text spotting can be further improved, *i.e.*, ID_{F_1}, MOTA and MOTP boosted by 0.2%, 0.4% and 0.2%, respectively. A possible reason for the performance improvement is that, when texts of temporal-adjacent frames are taken into account to perform semantic reasoning, some occluded texts can be correctly predicted since they may not be occluded in the adjacent frames.

Temporal Info.	End-to-End Video Text Spotting (%)			
	ID_{F_1}	MOTA	MOTP	FPS
✗	72.8	60.1	76.4	**15.9**
✓	**73.0**	**60.5**	**76.6**	10.8

Table 5: End-to-end text spotting performance of VLSpotter via semantic reasoning with or without texts from temporal-adjacent frames. The experiments are conducted on ICDAR 2015 Video.

5 Conclusion

This paper proposes an accurate video text spotter, VLSpotter, which leverages the semantic relationships among texts within a video frame to correct those wrongly spelled texts from the vision model of the text recognizer. To model the inter-text semantic relationships, we propose a text-wise semantic reasoning module to perform semantic reasoning for mitigating wrongly spelled text predictions. Moreover, we improve VTS performance by proposing a pluggable text-focused super-resolution module along with a mask-adaptive super-resolution loss that significantly alleviates motion blurs and enhances video quality. The experimental results demonstrate that VLSpotter outperforms the state-of-the-art methods by a clear margin in end-to-end video text spotting.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No.62176060), STCSM projects (No.20511100400, No.22511105000), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103), Shanghai Research and Innovation Functional Program (No.17DZ2260900), and the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

References

[Bhunia *et al.*, 2021] Ayan Kumar Bhunia, Aneeshan Sain, Amandeep Kumar, Shuvozit Ghose, Pinaki Nath Chowdhury, and Yi-Zhe Song. Joint visual semantic reasoning: Multi-stage decoder for text recognition. In *ICCV*, pages 14940–14949, 2021.

[Chen *et al.*, 2021a] Jingye Chen, Bin Li, and Xiangyang Xue. Scene text telescope: Text-focused scene image super-resolution. In *CVPR*, pages 12026–12035, 2021.

[Chen *et al.*, 2021b] Jingye Chen, Bin Li, and Xiangyang Xue. Zero-shot Chinese character recognition with stroke-level decomposition. In *IJCAI*, pages 615–621, 2021.

[Chen *et al.*, 2022a] Jingye Chen, Haiyang Yu, Jianqi Ma, Bin Li, and Xiangyang Xue. Text gestalt: Stroke-aware scene text image super-resolution. In *AAAI*, volume 36, pages 285–293, 2022.

[Chen *et al.*, 2022b] Xiangyu Chen, Xintao Wang, Jiantao Zhou, and Chao Dong. Activating more pixels in image super-resolution Transformer. *arXiv preprint arXiv:2205.04437*, 2022.

[Cheng *et al.*, 2019] Zhanzhan Cheng, Jing Lu, Yi Niu, Shiliang Pu, Fei Wu, and Shuigeng Zhou. You only recognize once: Towards fast video text spotting. In *ACM MM*, pages 855–863, 2019.

[Cheng *et al.*, 2020] Zhanzhan Cheng, Jing Lu, Baorui Zou, Liang Qiao, Yunlu Xu, Shiliang Pu, Yi Niu, Fei Wu, and Shuigeng Zhou. FREE: A fast and robust end-to-end video text spotter. *IEEE Transactions on Image Processing*, 30:822–837, 2020.

[Epshtein *et al.*, 2010] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR*, pages 2963–2970, 2010.

[Fang *et al.*, 2021] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *CVPR*, pages 7098–7107, 2021.

[Feng *et al.*, 2021] Wei Feng, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Semantic-aware video text detection. In *CVPR*, pages 1695–1705, 2021.

[He *et al.*, 2022] Yue He, Chen Chen, Jing Zhang, Juhua Liu, Fengxiang He, Chaoyue Wang, and Bo Du. Visual semantics allow for textual reasoning better in scene text recognition. In *AAAI*, volume 36, pages 888–896, 2022.

[Karatzas *et al.*, 2013] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. ICDAR 2013 robust reading competition. In *ICDAR*, pages 1484–1493, 2013.

[Karatzas *et al.*, 2015] 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 *ICDAR*, pages 1156–1160, 2015.

[Khare *et al.*, 2017] Vijeta Khare, Palaiahnakote Shivakumara, Raveendran Paramesran, and Michael Blumenstein. Arbitrarily-oriented multi-lingual text detection in video. *Multimedia Tools and Applications*, 76(15):16625–16655, 2017.

[Koo and Kim, 2013] Hyung Il Koo and Duck Hoon Kim. Scene text detection via connected component clustering and nontext filtering. *IEEE Transactions on Image Processing*, 22(6):2296–2305, 2013.

[Kuhn, 1955] Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[Li *et al.*, 2021] Zhuang Li, Weijia Wu, Mike Zheng Shou, Jiahong Li, Size Li, Zhongyuan Wang, and Hong Zhou. Contrastive learning of semantic and visual representations for text tracking. *arXiv preprint arXiv:2112.14976*, 2021.

[Liao *et al.*, 2020] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *AAAI*, volume 34, pages 11474–11481, 2020.

[Ma *et al.*, 2023] Jianqi Ma, Shi Guo, and Lei Zhang. Text prior guided scene text image super-resolution. *IEEE Transactions on Image Processing*, 32:1341–1353, 2023.

[Mou *et al.*, 2020] Yongqiang Mou, Lei Tan, Hui Yang, Jingying Chen, Leyuan Liu, Rui Yan, and Yaohong Huang. Plugnet: Degradation aware scene text recognition supervised by a pluggable super-resolution unit. In *ECCV*, pages 158–174, 2020.

[Shashirangana *et al.*, 2020] Jithmi Shashirangana, Heshan Padmasiri, Dulani Meedeniya, and Charith Perera. Automated license plate recognition: A survey on methods and techniques. *IEEE Access*, 9:11203–11225, 2020.

[Shivakumara *et al.*, 2017] Palaiahnakote Shivakumara, Liang Wu, Tong Lu, Chew Lim Tan, Michael Blumenstein, and Basavaraj S Anami. Fractals based multi-oriented text detection system for recognition in mobile video images. *Pattern Recognition*, 68:158–174, 2017.

[Spolaor *et al.*, 2020] Newton Spolaor, Huei Diana Lee, Weber Shoity Resende Takaki, Leandro Augusto Ensina, Claudio Saddy Rodrigues Coy, and Feng Chung Wu. A systematic review on content-based video retrieval. *Engineering Applications of Artificial Intelligence*, 90:103557, 2020.

[Stahlberg, 2020] Felix Stahlberg. Neural machine translation: A review. *Journal of Artificial Intelligence Research*, 69:343–418, 2020.

[Su *et al.*, 2023] Shangchao Su, Haiyang Yu, Bin Li, and Xiangyang Xue. Privacy-preserving collaborative Chinese text recognition with federated learning. *arXiv preprint arXiv:2305.05602*, 2023.

[Wang *et al.*, 2017] Xiaobing Wang, Yingying Jiang, Shuli Yang, Xiangyu Zhu, Wei Li, Pei Fu, Hua Wang, and Zhenbo Luo. End-to-end scene text recognition in videos based on multi frame tracking. In *ICDAR*, volume 1, pages 1255–1260, 2017.

[Wang *et al.*, 2018] Lan Wang, Yang Wang, Susu Shan, and Feng Su. Scene text detection and tracking in video with background cues. In *ICMR*, pages 160–168, 2018.

[Wang *et al.*, 2019] 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 *CVPR*, pages 9336–9345, 2019.

[Wang *et al.*, 2020] Wenjia Wang, Enze Xie, Xuebo Liu, Wenhai Wang, Ding Liang, Chunhua Shen, and Xiang Bai. Scene text image super-resolution in the wild. In *ECCV*, pages 650–666, 2020.

[Wang *et al.*, 2021] Wenhai Wang, Enze Xie, Xiang Li, Xuebo Liu, Ding Liang, Zhibo Yang, Tong Lu, and Chunhua Shen. PAN++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5349–5367, 2021.

[Wu *et al.*, 2015] Liang Wu, Palaiahnakote Shivakumara, Tong Lu, and Chew Lim Tan. A new technique for multi-oriented scene text line detection and tracking in video. *IEEE Transactions on Multimedia*, 17(8):1137–1152, 2015.

[Wu *et al.*, 2021] Weijia Wu, Yuanqiang Cai, Debing Zhang, Sibo Wang, Zhuang Li, Jiahong Li, Yejun Tang, and Hong Zhou. A bilingual, openworld video text dataset and end-to-end video text spotter with Transformer. *arXiv preprint arXiv:2112.04888*, 2021.

[Wu *et al.*, 2022] Wejia Wu, Zhuang Li, Jiahong Li, Chunhua Shen, Hong Zhou, Size Li, Zhongyuan Wang, and Ping Luo. Real-time end-to-end video text spotter with contrastive representation learning. *arXiv preprint arXiv:2207.08417*, 2022.

[Xu *et al.*, 2018] Yan Xu, Siyuan Shan, Ziming Qiu, Zhipeng Jia, Zhengyang Shen, Yipei Wang, Mengfei Shi, I Eric, and Chao Chang. End-to-end subtitle detection and recognition for videos in east asian languages via CNN ensemble. *Signal Processing: Image Communication*, 60:131–143, 2018.

[Xu *et al.*, 2021] Xingqian Xu, Zhifei Zhang, Zhaowen Wang, Brian Price, Zhonghao Wang, and Humphrey Shi. Rethinking text segmentation: A novel dataset and a text-specific refinement approach. In *CVPR*, pages 12045–12055, 2021.

[Yin *et al.*, 2013] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):970–983, 2013.

[Yu *et al.*, 2020] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, pages 12113–12122, 2020.

[Yu *et al.*, 2021a] Haiyang Yu, Jingye Chen, Bin Li, Jianqi Ma, Mengnan Guan, Xixi Xu, Xiaocong Wang, Shaobo Qu, and Xiangyang Xue. Benchmarking Chinese text recognition: Datasets, baselines, and an empirical study. *arXiv preprint arXiv:2112.15093*, 2021.

[Yu *et al.*, 2021b] Hongyuan Yu, Yan Huang, Lihong Pi, Chengquan Zhang, Xuan Li, and Liang Wang. End-to-end video text detection with online tracking. *Pattern Recognition*, 113:107791, 2021.

[Yu *et al.*, 2022] Haiyang Yu, Jingye Chen, Bin Li, and Xiangyang Xue. Chinese character recognition with radical-structured stroke trees. *arXiv preprint arXiv:2211.13518*, 2022.

[Zhao *et al.*, 2010] Xu Zhao, Kai-Hsiang Lin, Yun Fu, Yuxiao Hu, Yuncai Liu, and Thomas S Huang. Text from corners: a novel approach to detect text and caption in videos. *IEEE Transactions on Image Processing*, 20(3):790–799, 2010.

[Zhou *et al.*, 2017] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: An efficient and accurate scene text detector. In *CVPR*, pages 5551–5560, 2017.

[Zu *et al.*, 2022] Xinyan Zu, Haiyang Yu, Bin Li, and Xiangyang Xue. Chinese character recognition with augmented character profile matching. In *ACM MM*, pages 6094–6102, 2022.