FREE: A Fast and Robust End-to-End Video Text Spotter

Zhanzhan Cheng, Jing Lu, Baorui Zou, Liang Qiao, Yunlu Xu, Shiliang Pu, Yi Niu, Fei Wu, and Shuigeng Zhou

***Abstract*—Currently, video text spotting tasks usually fall into the four-staged pipeline: detecting text regions in individual images, recognizing localized text regions frame-wisely, tracking text streams and post-processing to generate final results. However, they may suffer from the huge computational cost as well as sub-optimal results due to the interferences of low-quality text and the none-trainable pipeline strategy. In this paper, we propose a fast and robust end-to-end video text spotting framework named FREE by only recognizing the localized text stream one-time instead of frame-wise recognition. Specifically, FREE first employs a well-designed spatial-temporal detector that learns text locations among video frames. Then a novel text recommender is developed to select the highest-quality text from text streams for recognizing. Here, the recommender is implemented by assembling text tracking, quality scoring and recognition into a trainable module. It not only avoids the interferences from the low-quality text but also dramatically speeds up the video text spotting. FREE unites the detector and recommender into a whole framework, and helps achieve global optimization. Besides, we collect a large scale video text dataset for promoting the video text spotting community, containing 100 videos from 21 real-life scenarios. Extensive experiments on public benchmarks show our method greatly speeds up the text spotting process, and also achieves the remarkable state-of-the-art.**

***Index Terms*—video text spotting, end-to-end, detection, tracking, quality scoring**

I. Introduction

VIDEO text spotting is still an important research topic due to its large amount of applications such as port container number identification in industrial monitoring, license plate recognition in the intelligent transportation system, and road sign recognition in advanced driver assistance system.

Previous methods [1], [2], [3] are usually fourfold: detecting text regions in individual images, recognizing localized text regions one-by-one, tracking text regions as streams, and applying post-processing techniques for generating the final results. However, these methods suffer from two major problems: 1) Massive computational overhead because of the one-by-one text recognition strategy, which may be impracticable especially when working on front-end devices such as surveillance cameras or even in-vehicle cameras. 2) Sub-optimal results due to the overwhelming of *low-quality* (*e.g.* blurring, perspective distortion, rotation and poor illumination, etc.) text and none-trainable pipeline strategies. In practice, it is unnecessary to recognize each text region in text streams, which will result in huge computational costs and also bring various interferences of low-quality text. Learned from some end-to-end text spotting methods [4], [5], [6], [7] in single images, the non-trainable pipeline strategies will also decrease video text spotting performance. Besides, some motion interferences (*e.g.* object/camera moving or shaking) will lead to text regions missing in video text detection, and then the detectors will suffer from the low recall problem, as shown in previous methods [8], [9], [10], [11], [12], [13].

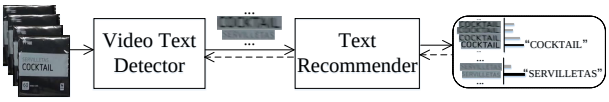


Fig. 1. Illustration of the proposed FREE, which consists of two sub-modules: the video text detector for generating candidate text regions and the text recommender for recognizing the highest-quality text regions in their tracked text streams. Dashed lines denote back-propagation.

To circumvent the above problems, the main idea is to select the *highest-quality* (*e.g.* clear and horizontal) text region from each text stream, and then only the selected text region needs to be recognized. Thus the selection-and-recognition strategy is much more efficient than those one-by-one strategies. Correspondingly, it needs us to design a robust quality scorer to assign a quality score to each detected text. To further speed up the process of video text spotting, we also attempt to simplify and assemble the text tracking, text quality scoring and recognition into a unified trainable module named text recommender (See 'text recommender' in Figure 1). In this way, text recommender will benefit from the complementarity among tracking, scoring and recognition. It not only largely decreases the interferences of low-quality text, and also greatly decreases the computational cost compared to the one-by-one recognition strategy. For the low recall problem in detection, spatial-temporal information (*e.g.* text location and context) between consecutive video frames can be exploited to recall text regions as many as possible, which is important to generate complete text streams. Finally, we can integrate the spatial-temporal video text detector and the text recommender into a framework, and train the framework end-to-end for

Z. Cheng and F. Wu are with College of Computer Science and Technology, Zhejiang University, Hangzhou, 310058, China (e-mail: 11821104@zju.edu.cn, wufei@cs.zju.edu.cn). Z. Cheng is also with Hikvision Research Institute, Hangzhou, 310051, China.

J. Lu, L. Qiao, Y. Xu, S. Pu and Y. Niu are with Hikvision Research Institute, Hangzhou, 310051, China (email: lujing6@hikvision.com, qiaoliang6@hikvision.com, xuyunlu@hikvision.com, pushiliang.hri@hikvision.com, niuyi@hikvision.com).

B. Zou, S. Zhou are with Shanghai Key Lab of Intelligent Information Processing and School of Computer Science, Fudan University, Shanghai, 201203, China (email: 18210240270@fudan.edu.cn, sgzhou@fudan.edu.cn).

Both Z. Cheng and J. Lu contributed equally to this research. S. Pu is the Corresponding author.

Manuscript received April xx, 20xx; revised August xx, 20xx.

global optimization.

In this paper, we propose a **F**ast and **R**obust **E**nd-to-**E**nd video text spotting approach named as FREE by integrating a well-designed spatial-temporal video text detector and a text recommender into an end-to-end trainable framework, as shown in Figure 1. Concretely, the spatial-temporal video text detector is designed to recall more text by referring to the temporal relationship between consecutive video frames. Text recommender assembles text tracking, text quality scoring and text recognition into a unified trainable network. Here, the tracking module is responsible for generating text streams, and the quality scoring module is designed to assess the quality of each text region. Then the detector and the text recommender are combined as a network, which can be trained in an end-to-end manner (detailed in *Methodology Section*). Note that, the proposed framework is theoretical much faster than existing multi-stage methods [1], [2], [3], [14]. This is because FREE just needs to recognize the one and only one text region with the highest quality score in a tracked text stream, contrasting to previous methods recognizing every text region from the tracked text stream. As a result, it can largely speed up the process of the recognition process, progressively improve the efficiency of the video text spotting.

Last but not least, we also note that the scenario scales of existing video text benchmarks are limited. For example, the largest video scene text dataset 'Text in Videos' [15] only has 49 videos from 7 different scenarios, which may limit researches on video text understanding. In this paper, we collect a **l**arge-**s**cale **v**ideo **t**ext **d**ataset (*abbr*. LSVTD) containing 100 videos from **21** natural scenarios, and hope to help the research of video text understanding.

Contributions of this paper are summarized as follows: (1) We design a novel text recommender for selecting the highest-quality text from text streams and then only recognizing the selected text regions once. It significantly speeds up the recognition process, and also improves the video text spotting performance. (2) We integrate a well-designed spatial-temporal text detector and a text recommender into an end-to-end trainable framework named as FREE for fast and robust spotting video text. The spatial-temporal detector can help mine more text regions between consecutive frames. (3) To promote the progress of video text spotting, we collect and annotate a larger scale video text dataset, which contains 100 videos from 21 different real-life situations. (4) Extensive experiments demonstrate that our method is fast and robust and achieves impressive performance in video scene text spotting.

Declaration of major extensions compared to the conference version [8]: (1) We achieve the video text spotting in an end-to-end trainable manner instead of the two-staged form in its conference version. To achieve this, we replace EAST [16] with an end-to-end trainable text spotting framework Text Perceptron [17] (*abbr.* TP), in which the original recognition module in TP is replaced with our text recommender sub-module. (2) We further enhance the text recommender module by redesigning the template estimation mechanism in a learnable manner rather than roughly synthesizing templates by K-Means. This is because K-Means is inherently sensitive to outlier samples and not robust to complex scenarios. (3) Correspondingly, we explore the effects of FREE with more extensive experimental evaluations, which demonstrates the advantages of the extended version. Besides, we refine LSVTD by removing some consecutive background frames, and provide more detailed characteristics.

II. Related work

With the rapid development of artificial intelligence techniques [18], [19], [20], [21], great progress has been made in many isolated applications such as causal inference [22], named entities identification [23], question answering [24], scene text spotting [5], [6], [17] and video understanding [25], [26]. However, it is very important to build multiple knowledge representation [27] for understanding the real and complex world. Real-time text spotting [2], [3], [8] is such a complex task helping to understand the complex world, which actually needs heterogeneous technique fusion including object detection, tracking as well as scene text recognition techniques. Here, we roughly brief the text spotting techniques into two categories as follows.

A. Text Reading in Single Images

Traditionally, the scene text reading system contains a text detector for localizing each text region and a text recognizer for generating corresponding character sequences. For text detection, numerous methods are proposed to localize regular and irregular (oriented and curved etc.) text regions, which can be categorized as anchor-based [28], [29], [30], [31], [32], [33] and direct-regression-based [34], [35], [16]. For text recognition, the task is now treated as a sequence recognition problem, in which CTC [36]-based [37], [38], [39] and attention-based [40], [41], [42], [43], [44] methods are designed and have achieved promising results.

Recently, in order to sufficiently exploit the complementarity between detection and recognition, many methods [45], [4], [5], [6], [46], [7], [17], [47], [37], [48], [49], [50] are proposed to spot text in an end-to-end manner, which utilize the recognition information to optimize the localization task.

In fact, lots of text reading applications actually work in video scenarios, in which scene text spotting from multiple frames may be more meaningful.

B. Text Reading in Videos

In recent years, only a few attention has been drawn to spotting video scene text in contrast to text reading in still images. For more details of text detection, tracking and recognition in video, the readers can refer to a comprehensive survey [51]. In general, reading text from scene videos can be roughly categorized into three major modules: 1) text detection, 2) text tracking, and 3) text recognition.

Text detection in videos. In the early years (before 2012), most methods focus on detecting text in each frame with connected component analysis [9] or sliding window [52] strategy. However, the performance of them is limited due to the low representation of handcrafted features. Though the recent detection techniques (mentioned in *Section*. 2.1) in still

images can help improve feature representation, detecting text in scene videos is still challenging because of its complicated temporal characteristics (*e.g.* motion). Therefore, text tracking strategies are introduced for enhancing the detection performance, which are further divided into two categories [51]: spatial-temporal information based methods [53], [54], [55], [56] for reducing noise and fusion based methods [57], [58], [59], [60] for improving detection accuracy. Recently, Yang et al. [61] proposed a tracking based multi-orientation scene text detection method using multiple frames within a unified framework via dynamic programming. Khare et al. [62] introduced the automatic windows to extract moments for tackling multi-font and multi-sized text in the video. Shivakumara et al. [63] introduced fractals for enhancing text detection in videos, especially in the low-resolution mobile video. Wang et al. [10] employed an optical flow-based method to refine text locations in the subsequent frames. Wang et al. [64] proposed a video text detection network, which combines complementary text features from multiple related frames to enhance the overall detection performance. Wang et al. [65] proposed a fully convolutional network model for detecting text in videos based on a defined refine block structure.

Text tracking in videos. The traditional methods such as template matching [55], [54], [66], [67] and particle filtering were popular. But these methods failed to solve the re-initialization problem, especially in scene videos. Then the tracking-by-detection based methods [2], [68], [69] were developed to estimate the tracking trajectories and solve this problem.

Recently, Zuo et al. [59] and Tian et al. [70] attempted to fuse multi-tracking strategies (*e.g.* spatial-temporal context learning [71], tracking-by-detection, etc.) for text tracking, in which the Hungarian [72] algorithm was applied for generating the final text streams. Wu et al. [73] proposed a technique for detecting and tracking video text of any orientation by using spatial and temporal information, respectively. Yang et al. [74] also proposed a motion-based tracking approach in which detected results are directly propagated to the neighboring frames for recovering missing text regions. Wang et al. [75] presented a scene text detection and tracking method for videos, in which the enhanced EAST model by de-convolution layers and the correlation filter based tracking algorithm were developed to improve the detection and tracking results. Wang et al. [76] proposed a new video text tracking approach based on hybrid deep text detection and layout constraint. Yu et al. [77] proposed an end-to-end video text detection model with online tracking to address video text detection and tracking challenges. In fact, the robust feature extractor is the most important component of a text tracker.

Text recognition in videos. With the tracked text streams, there are two strategies for better scene text recognition: selection strategy by selecting the best text regions from streams (popular before 2010), and results fusion strategy by combining corresponding recognized character results. Correspondingly, methods [55], [54], [56] selected the region with the longest horizontal length as the most appropriate region. Then Goto and Tanaka [53] further enhanced the selection algorithm by taking six different features (*e.g.* Fisher's discriminant ratio, text region area, etc.) into account. While recent methods [69], [78] directly fused recognized results in text streams for final text predictions by majority voting, CRF or frame-wise comparison, and these approaches assumed that recognition results in most frames are trust-worthy, which may not be true in unconstrained scenarios. In addition, frame-wise text recognition also results in a high computation cost.

End-to-end text recognition in videos. There are several works proposed to solve the end-to-end video text spotting problem. Nguyen et al. [2] first proposed an end-to-end video text reading solution by extending Wangs's method [14], in which the frame-wise detection and the tracking with multiple features (*e.g.* the temporal distance, edit distance, etc.) are applied. Merino-Gracia and Mirmehdi [1] proposed an end-to-end video scene text reading system by introducing the unscented Kalman filter [79], but mainly focused on large text found in outdoor environments. Recently, Wang et al. [3] firstly utilized an end-to-end deep neural network to detect and recognize text in each frame, and then employed the *tracking-by-detection* strategy to associate text regions, and recovered missed detections with the tracked results, finally obtained the recognition results by voting the most frequently appeared text strings.

In fact, it is a very challenging task to optimize video text spotter end-to-end when taking multiple functional modules (text detection, text tracking and text recognition) into consideration, especially compared to the traditional four-staged pipeline strategy. Therefore, in this paper we develop an end-to-end trainable video text spotter with only two trainable modules: the video text detector and the text recommender, similar to the end-to-end text spotting methods [6], [17], [45], [47], [48], [49] in single images.

III. Methodology

The proposed method consists of two parts, the spatial-temporal video text detector and the text recommender, as shown in Figure 2. In the following sections, we first describe the two parts in Section III-A and III-B respectively. Then, we describe the training and inference strategy in Section III-C.

A. Spatial-Temporal Video Text Detector

1) Text Detection in Single Frame: Since TP [17] is a more robust text spotter than EAST [16] especially on irregular text detection, and also can be trained end-to-end. We redesign and implement the video text detector inspired by the TP architecture (including a text detection module, a shape transform module and a recognition module), as shown in Figure 2. Here, the detection backbone and text shape transform operation are the same as the detection part and shape transform module in TP, respectively.

2) Text Detection in Consecutive Frames: Considering the spatial-temporal characteristic in the video, we learn relations between consecutive frames with a *spatial-temporal aggregation* strategy for improving video text detection, which is divided into three steps: 1) enhancing temporal coherence between frames with a feature warping mechanism [80], 2) spatial matching between frames with a comparing and

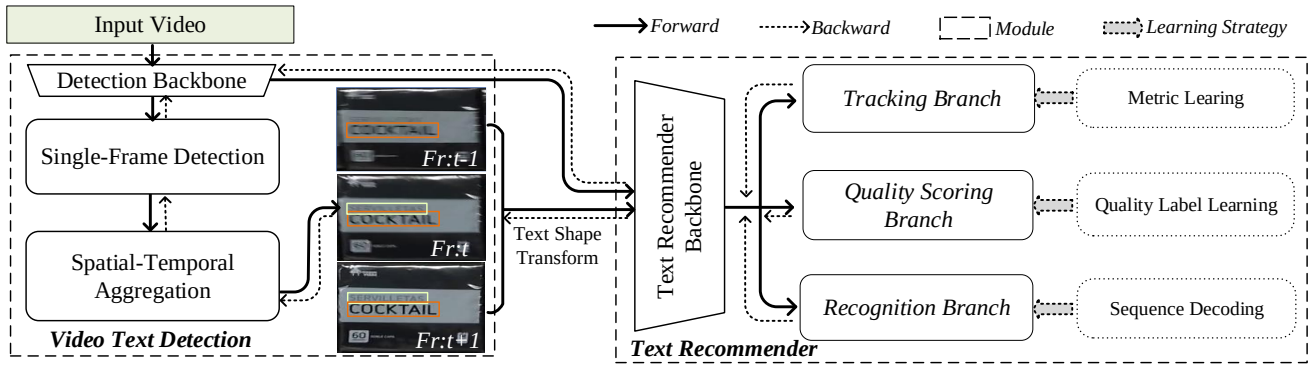


Fig. 2. The architecture of FREE, which consists of two parts: (a) The spatial-temporal text detector for generating text regions; (b) The text recommender assembling the tracking, quality scoring and recognition into a unified trainable network. Specifically, the tracking, quality scoring and recognition branches are simultaneously learned with the metric learning, the quality label learning and the sequence decoding, respectively. Fr:(.) means the frame ID.

matching strategy inspired by [81], [82], and 3) temporal aggregation.

Formally, let I_t be the t-th frame in a video, the detection results in I_t can be refined with the detections of its consecutive frames $(I_{t-n}, ..., I_{t+n})$ where the size of the refining window is $2n$+1.

Enhancing temporal coherence. We obtain the corresponding sequence of feature maps F=$(F_{t-n}, ..., F_{t+n})$ by propagating frames through the detection backbone. Given a pair of frame features F_{t+i} and F_t (the reference frame), we enhance their temporal coherence by referring to the estimated flow $flow_{(t+i,t)}$ between I_{t+i} and I_t with a flow-guided warping mechanism

$$F^w_{t+i} = Warp(F_{t+i}, flow_{(t+i,t)}), \tag{1}$$

where $flow_{(t+i,t)}$ is pre-computed with TV-L1 algorithm, $Warp(.)$ is the bilinear warping function applied on each element in the feature maps, and F^w_{t+i} denotes the feature maps warped from frame I_{t+i} to frame I_t. Thus F is further transferred as the warped $F^w = (F^w_{t-n}, ..., F^w_{t+n})$. Then we generate an enhanced sequence of *confidence maps* $C = (C_{t-n}, ..., C_{t+n})$ by propagating F^w into a classification subnetwork, in which each value in C_{t+i} represents the possibility of being a text region.

Comparing and matching. We evaluate the spatial matching degrees of two frames with matching weights. The weights are firstly computed with a transform module to produce the feature-aware filter which is represented as

$$F^{trans}_{t+i} = ReLU(BN(WF^w_{t+i} + b)), \tag{2}$$

where W and b are learnable parameters, BN and ReLU represent Batch Normalization and rectified linear unit function, respectively. Given the transformed feature maps, we compute the similarity energy $Sim_{t+i,t} = F^{trans}_{t+i} \cdot F^{trans}_t$ of I_{t+i} and I_t as the matching weights, where '$\cdot$' means the dot product.

Temporal aggregation. We compute the aggregation weights by

$$a_{t+i} = \frac{exp(Sim_{t+i,t} \odot C_{t+i})}{\sum^n_{i'=-n} exp(Sim_{t+i',t} \odot C_{t+i'})}, \tag{3}$$

where '$\odot$' represents the element-wise product. Here, we multiply $Sim_{t+i,t}$ by C_{t+i} to reinforce the aggregation weights of positive detections. Then the temporal aggregation across the consecutive frames is computed by

$$C_{t,agg} = \sum^n_{i=-n} a_{t+i} \odot C_{t+i}. \tag{4}$$

To handle few mis-aggregated situations, we further refine $C_{t,agg}$ as $C_{t,ref} = C_{t,agg} \odot M_t$ by applying a normalized binary mask M_t to $C_{t,agg}$, where M_t is calculated by normalizing F^w_t as a binary mask with a pre-set threshold (default by 0.5).

To better declare the spatial-temporal aggregation mechanism, we describe it in Algorithm 1. Here, the refining window-size is set to 5.

Note that, we conduct aggregation operation on C instead of F^w because the feature maps are later fed into the regression branch to determine the geometry shapes of text regions. Since the geometry shapes of the same text usually vary in different frames, the values in F^w to regress text positions are also very different. Therefore, if we directly aggregate F^w, the feature values to regress different geometry shapes will be mistakenly merged and impact the final regression result of I_t. On the contrary, confidence maps have no such problem because they only indicate the possibility of text existence.

With the aggregated results, the optimization is with three tasks [17], i.e., the binary Dices Coefficient Loss [83] (denoted by $\mathcal{L}_{cls}$) to learn text boundaries and its central regions, the Corner Regression loss (denoted by $\mathcal{L}_{corner}$) to regress the offsets to their corner points, and the Boundary Offset Regression loss (denoted by $\mathcal{L}_{boundary}$) to regress the vertical and horizontal offsets to their nearest boundaries. The loss function is as follow:

$$\mathcal{L}_D = \mathcal{L}_{cls} + \lambda_b \mathcal{L}_{corner} + \lambda_c \mathcal{L}_{boundary}, \tag{5}$$

where λ_b and λ_c are auto-tunable parameters.

B. Text Recommender

The text recommender contains three trainable subparts: quality scoring, tracking and recognition.

Algorithm 1 The spatial-temporal aggregation process.

$N = len(Frames)$; %length of video frames
init buffer memory for feature and confidence maps to avoid repetitive computation: fb,cb with $size = 2n + 1$;
for each $i \in [1, N]$ **do**
  $start = max(1, i - n)$;
  $end = min(N, i + n)$;
  $idx_r = i - start + 1$;
  if $fb[idx_r]$ is empty **then**
    $fb[idx_r] = DetNet(I_i)$;
    $cb[idx_r] = Cls(fb[idx_r])$;
  end if
  $weight = cb[idx_r]$; %init weight sum
  $conf = cb[idx_r] \odot cb[idx_r]$; %init refined confmap
  for each $j \in [start, end]$ **do**
    $idx_n = j - start + 1$;
    if idx_n==idx_r **then**
      continue;
    end if
    if $fb[idx_n]$ is empty **then**
      $fb[idx_n] = DetNet(I_j)$;
      $cb[idx_n] = Cls(fb[idx_n])$;
    end if
    $F_j^w = Warp(fb[idx_n], flow_{(j,i)})$;
    $C_j = Warp(cb[idx_n], flow_{(j,i)})$;
    $W(j, i) = exp((F_j^w \cdot fb[idx_r) \odot C_j)$
    $weight = weight + Wj, i$;
    $conf = conf + W(j, i) \odot C_j$;
  end for
  $M(i) = Threshold(fb[idx_r])$; %binary mask
  $conf = M(i) \odot conf/weight$; %final confmap
  get detection result with $fb[idx_r]$ and $conf$;
  for $i = 2$ to $2n + 1$ **do**
    $fb[i - 1] = fb[i]$; %updating fb for next frame
    $cb[i - 1] = cb[i]$; %updating cb for next frame
  end for
  set $fb[2n + 1]$ empty;
  set $cb[2n + 1]$ empty;
end for

1) Text Quality Scoring: It is almost impossible to manually annotate continuous quality scores (*e.g.* ranging from 0.0 to 1.0) for text regions because the judgment of imaging quality is subjective for different annotators. It means that annotators cannot definitely and quantitatively rank the quality of text regions with different interferences such as shape deformation, blur and even occlusion, etc. Besides, the annotating cost is also very tremendous. We empirically find features of different quality images are actually localized at different positions in feature space, which implies the quality correlations are contained in feature distributions intrinsically. Here we propose the concept of the *standard template* that is assumed as the feature representation of an ideal text image without any interference, i.e., representing the highest quality. Naturally, the represented template always can be recognized correctly, and also should be close to those high-quality images while being far from the low-quality ones. This is based on the common phenomena that higher-quality images usually have a better chance for accurate recognitions, the images that are close to their template are the ones leading to correct recognitions, vice versa. In this way, we can define the feature distance between the template and a candidate text region as the corresponding quality score.

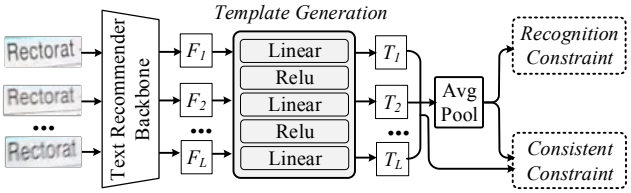


Fig. 3. The illustration of template generation network.

The template generation is the key to generate quality scores. We here design the template generation strategy in a weakly-supervised fashion, as shown in Figure 3. Then we obtain quality labels by computing the similarities between images and their templates. Thus the quality scoring branch can be optimized with the generated quality labels. The above process is detailed as follows.

Learning template. Since we assume that each text image is corresponding to a standard template image in the feature space, and the standard template should be closer to CORRECTLY recognized images. Hence, we can attempt to build the feature mapping from correctly recognized images to the standard template with neural networks such as a three-layer Linear module. The template learning process can be implemented with two constraints: 1) Naturally, the standard template should be beneficial for recognition as it represents the highest quality. 2) Text images from the same text stream should correspond to the same quality standard, thus the generated templates from the same stream should remain consistent. Specifically, the second constraint ensures that templates from the same stream are almost identical, thus we could conduct average pooling on them to obtain the final feature representation of the only one standard template. Note that, average pooling is a widely used method for joint representation learning, as used in most existing methods [84], [85], [86].

Given a text stream with L images that are correctly recognized (by the pre-trained recognition model, detailed in *Implementation Details*), we extract their features as $F = [F_1, F_2, ..., F_L]$ with the pre-trained recognition model. We generate the templates from a text stream as $T = [T_1, T_2, ..., T_L]$ with a three-layer Linear module. Then we integrate the generated templates into the final template representation $\bar{T}$ with average pooling.

To learn the representative template, we can use the text recognition loss (denoted by $\mathcal{L}_{recog}$) to optimize the synthesized template's feature so that it is beneficial for recognition. Since the quality scores of images in each text stream should be computed with respect to the same quality standard, a template consistency loss (denoted by $\mathcal{L}_{consis}$) is designed to

constrain the template's feature of each image being consistent. The loss function is formalized as:

$$\mathcal{L}_Q=\mathcal{L}_{recog}+\lambda_d\mathcal{L}_{consis} \tag{6}$$

where λ_d is a tunable parameter and the consistency loss is formalized as:

$$\mathcal{L}_{consis}=\frac{1}{L}\sum_{i=1}^{L}mse(T_i,\bar{T}) \tag{7}$$

in which mse is the average mean square error. T_i means the learned template of image F_i.

Generating quality labels. For each image, we can generate its quality label simply by computing the similarity between its feature F_i and its template:

$$s_i=\frac{\bar{T}\cdot F_i}{||\bar{T}||\cdot||F_i||}. \tag{8}$$

Optimizing quality scoring branch. Following 'Text Recommender Backbone' in Figure 2, the quality scoring branch consists of an attention decoder (shared with *recognition branch*, detailed in *Implementation Details*) for extracting features, followed by two Linear layers and a Sigmoid activation layer for regressing the quality scores. Supervised by the generated quality labels, the quality scoring branch can directly regress the quality score for each text region, as shown in Figure 2. Then the loss function for the quality scoring branch is as follows:

$$\mathcal{L}_S=\frac{1}{N}\sum_{i=0}^{N}||s_i\text{-}s_i'|| \tag{9}$$

where s' is the predicted quality score.

2) Text tracking: The tracking task aims to group corresponding text regions into text streams. Intuitively, the tracker should have the ability to ensure that the feature of a text region in one stream must remain closer distance to those in the same stream than others, which implies: 1) the features must be discriminative enough to tolerate various interferences in unconstrained scenes, and 2) the module may be better if trained with a good distance measure.

Robust feature extraction. Thanks to the studies in deep neural network and metric learning, we extract robust features for the tracker by applying the metric learning technique. Concretely, we firstly select three regions from localized candidate regions as an image triplet (R^a, R^p, R^n), in which R^a and R^p are corresponding to the same text instance while R^n is randomly selected from other text instances. We separately name R^a, R^p and R^n as the anchor, positive and negative samples. Secondly, an image triplet is fed into a deep CNN for generating its L2 Normalized high-level representation (q^a, q^p, q^n). The tracker is then trained with two metric learning loss: triplet loss [87] $\mathcal{L}_{triplet}=max(d(q^a,q^p)-d(q^a,q^n)+margin,0)$ and contrastive loss [88] $\mathcal{L}_{contra}=||q^a-q^p||$, i.e.,

$$\mathcal{L}_T=\mathcal{L}_{contra}+\lambda_t\mathcal{L}_{triplet}, \tag{10}$$

where d represents Euclidean distance, and λ_t and $margin$ are tunable parameters shown in *Implementation Details*. Here, triplet loss is responsible for distinguishing positive and negative samples, which only constrains that inter-class distance should be larger intra-class distance. While the contrastive loss is used for reducing the intra-class distance to better match text regions in a text stream. The two losses are complementary to each other for optimizing the tracking branch to make the network converge steadily.

Text stream generation. With the trained tracking model, for a pair of candidate text regions (R^1, R^2), we calculate its matching cost by

$$MC(R^1,R^2)=\frac{1}{q^1\odot q^2+\epsilon}. \tag{11}$$

To avoid division by zero error, ϵ is set as 10^{-7}. Then those pairs with MC larger than a threshold are considered as invalid matching pairs and filtered out. Finally, we employ the Hungarian algorithm [72] to generate the text streams.

3) Text Recognition: The text recognition module is not our focus, and we select the attention-based method as our decoder (shared with the *quality scoring branch*) just like in [41], [43], [44].

Since the above three submodules are complementary for extracting discriminative features, we jointly train all the three subtasks with same text recommender backbone in a unified trainable network, which is supervised by:

$$\mathcal{L}_{TR}=\lambda_1\mathcal{L}_T+\lambda_2\mathcal{L}_S+\lambda_3\mathcal{L}_R, \tag{12}$$

where $\mathcal{L}_R$ is the recognition part loss and λ_i (i=1,2,3) denotes the loss weight for different tasks.

C. Training and Inferences

1) Loss Function: The loss of the whole framework contains two parts: the video text detection and the text recommender, that is,

$$\mathcal{L}=\mathcal{L}_D+\lambda\mathcal{L}_{TR}, \tag{13}$$

where λ is an auto-tunable parameter used in [17].

2) Optimization: It is challenging to train FREE end-to-end from scratch. In a more achievable way, we first pre-train TP on static images with the same training strategy to that in [17]. Second, we equip the pre-trained TP with the spatial-temporal aggregation module and text recommender (including tracking and scoring and recognition) as FREE, and train the video detector and recommender separately, then fine-tune the whole framework in an end-to-end trainable way. Note that, it is easy to train the text recommender because the three branches (tracking, quality scoring and recognition) are complementary to each other. Thus FREE can be end-to-end trained by jointly optimizing the video text detector and recommender, just like the end-to-end text spotting by optimizing the text detector and recognizer in single images. Besides, the 'Text Recommender Backbone' and 'Recognition Branch' are the same as those in the pre-trained TP.

3) Inference Process: The inference process is divided into three steps, as shown in Figure 4: localizing texts from multiple frames, tracking candidate text regions into text streams, each of which is accompanied by the quality score, and recognizing the highest-quality text as a character sequence.

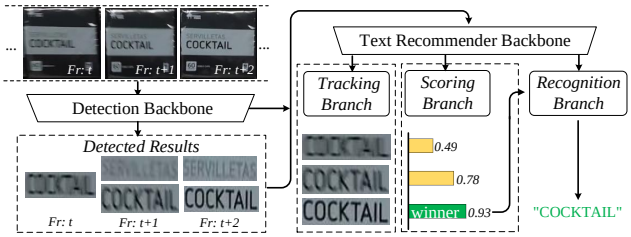


Fig. 4. The illustration of inferences.

IV. The Large-Scale Video Text Dataset

In recent years, research in video scene text spotting still remains unpopular in contrast to its promising application prospects. The limited video text dataset may be a major reason that restrains researches in this area. For example, existing video scene text datasets such as *IC13* [89] or *IC15* [15] (See Table I[1]) are limited on the scale of video items and scenarios. Though the recently released RoadText-1K (*abbr*. RT-1K) provides 1000 videos, it focuses on the road scenario. Thereby, we collect and annotate a **l**arge-**s**cale **v**ideo **t**ext **d**ataset (denoted by LSVTD). LSVTD has 100 scene videos consisting of 65615 frames and 563444 text instances, which is collected from 21 typical real-life scenarios (*train watch*, *city road*, *inside train*, *harbor surveillance*, *highway*, *inside shops*, *office building*, *outdoor shopping mall*, *bookstore*, *indoor shopping mall*, *bus/railway station*, *fingerpost*, *restaurant*, *pedestrian*, *hotel*, *shopping bags*, *digital screen*, *supermarket*, *street view*, *metro station* and *books opening*), illustrated in Figure 5. In the future, we will progressively increase the scale of this dataset.

TABLE I
Comparison with existing video text datasets. '*quality*' means if each text region is labeled with quality-level indication.

Datasets	#scenario	#video	#frame	#instance	quality
Merino [90]	4	–	–	–	
Minetto [58]	–	5	3,599	8,706	
IC13 [89]	7	28	15,277	93,934	✓
YVT [2]	–	30	13,500	–	
IC15 [15]	7	49	27,824	–	✓
RT-1K [91]	1	1,000	300,000	1,280,613	
LSVTD	21	100	69,577	535,065	✓

Dataset characteristics. LSVTD is detailed in Table I, and mainly characterized by 1) Much larger scale, which is more than twice the scale of IC15. 2) More diversified scenarios. LSVTD covers a wide range of 13 indoor (*e.g*. bookstore, shopping mall) and 8 outdoor (*e.g*. highway, city road) scenarios. The variety of scenarios challenges text spotting algorithms to achieve robust performance. 3) Multilingual text instances. LSVTD contains text with multiple languages (English and Chinese etc.) which are divided into 2 major categories: Latin and Non-Latin.

More concretely, we also list multiple characteristic attribute distributions on different scenarios, including *the number of videos*, *the number of frames*, *the number of text streams*, *the average number of text regions per frame*, and *the quality distribution* ('high','moderate' and 'low'), as shown in Figure 6. We find that LSVTD is very diverse. For example, the *outdoor shopping mall* is usually with a large amount of text while the *high way* is with less text (demonstrated in Figure 6.(d)), and *books opening* and *street view* are usually with more high-quality text than that in *city road* (demonstrated in Figure 6.(e)).

[1]The download link of YVT is broken now. We obtain this dataset from Yu et al. [77].

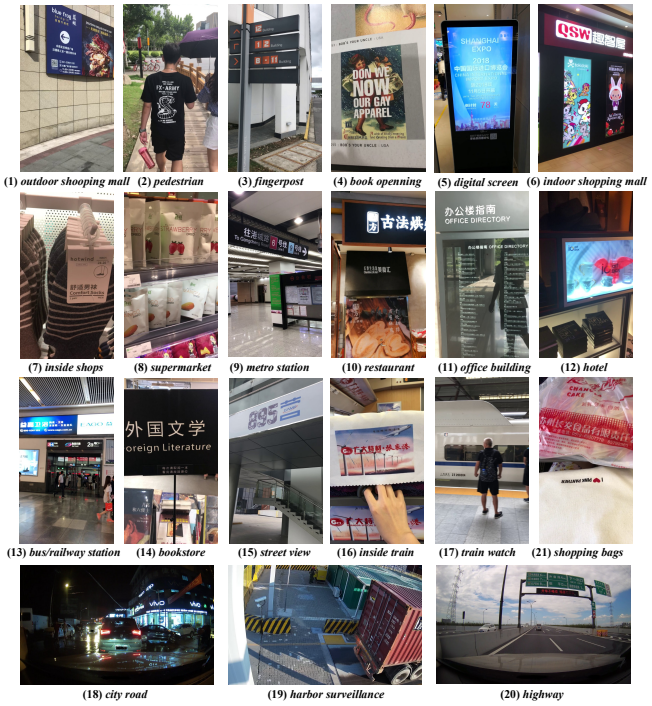
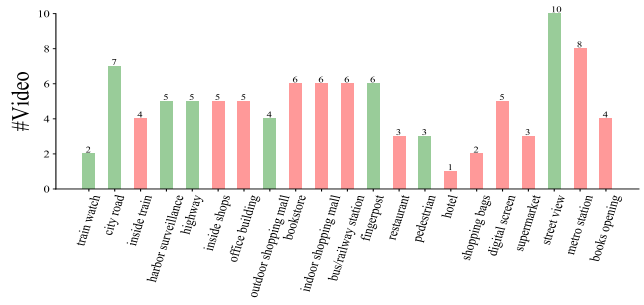


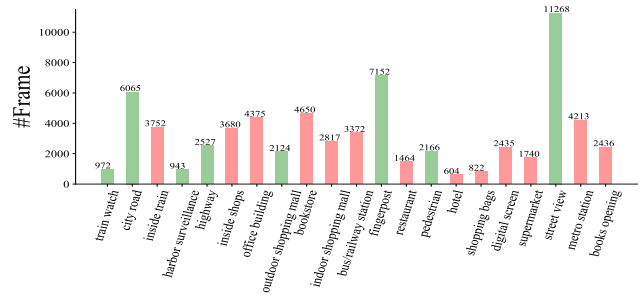
Fig. 5. Illustration of 21 different scenarios.

Annotation details. Following the annotation strategy in IC15, we annotate the following items for each text: 1) *Polygon coordinate* represents text location; 2) *ID* means the unique identification for each text among consecutive frames; That is, the same text in consecutive frames shares the same ID; 3) *Language* is categorized as Latin and Non-Latin as mentioned above; 4) *Quality* coarsely indicates the quality level of each text region, which can be qualitatively labeled as three quality levels: 'high' (recognizable, clear and without interferences), 'moderate' (recognizable but polluted with one or several interferences) or 'low' (one or more characters are unrecognizable). 5) *Transcripts* mean text string for each text region. Note that, though it is hard to annotate continuous quality scores, we still attempt to annotate the coarse quality-level indication of each text region for some algorithms' evaluation.

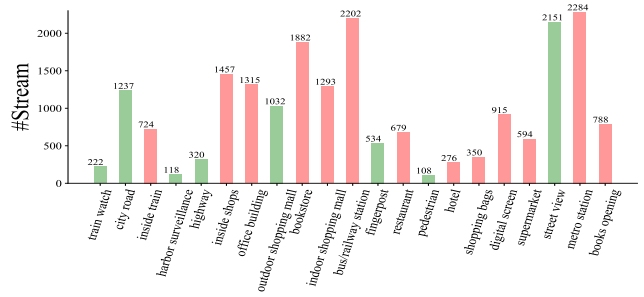
We parsed videos (ranging from 5 seconds to 1 minute) to frames and then instructed 6 experienced annotation workers to label them, and conducted cross-checking on each text region. It took more than 3500 man-hours for annotating these frames. We randomly select two-thirds of videos as the training videos (except for *hotel*), while the rest are treated as testing videos. Thereby, the training and testing datasets separately contain



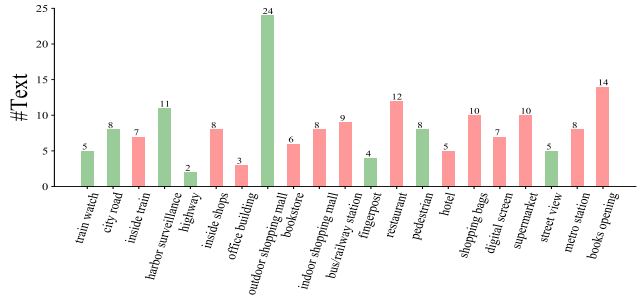
(a) The video number of each scenario.



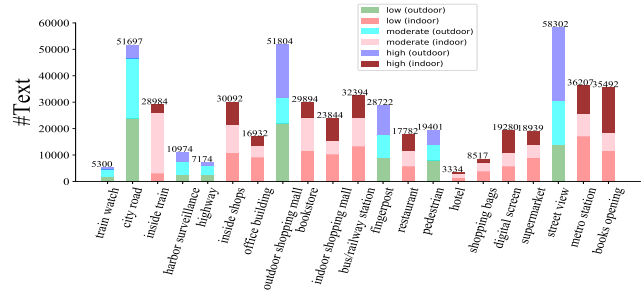
(b) The frame number of each scenario.



(c) The text stream number of each scenario.



(d) The average text instance number per frame in each scenario.



(e) Quality distribution of text in each scenario.

Fig. 6. Characteristic attribute distributions on different scenarios.

66 and 34 videos. Here, we hope that availability[2] of LSVTD may spur more interest in corresponding areas.

Differences of the conference version and the refined version. We refine LSVTD as follows: 1) removing the traffic surveillance scenario due to some privacy policies, 2) removing some segments of background frames without text content, and 3) adding 2, 1 and 2 videos for *city road*, *street view* and *bookstore* scenarios, respectively.

V. Evaluation Protocols

The classic evaluation protocols for text detection, tracking and recognition in videos have been declared in [51]. Since we only spot the highest-quality text instead of frame-wise recognition, we revisit and extend previous mainstream metrics necessarily for evaluating the effects of video text spotting.

A. Detection metrics

In video text spotting, a few methods explicitly evaluate the frame-wise detection performance without respecting to tracking or recognition results. Here, we also provide detection metrics solely for evaluating detection performance. Following detection methods [16], [29], [92] evaluated on single images, *precision* (*abbr.* PRE), *recall* (*abbr.* REC) and *F-measure* are used as the frame-wise detection metrics.

B. Tracking metrics

Tracking metrics should maximize the sum of the overlap between the tracking results and ground truth. In general, evaluation metrics [51], [93] like multiple object tracking precision (*abbr.* $MOTP$), multiple object tracking accuracy (*abbr.* $MOTA$), and the average tracking accuracy (*abbr.* ATA) are used to evaluate the performance of tracking. Notice that ATA is the most important metric because ATA measures the tracking performance over all the text, as addressed in [74].

C. Quality scoring metrics

In general, the better quality frames are selected, the higher accuracy we will get. To evaluate the performance of the quality scoring mechanism, we define the **q**uality **s**election **h**itting **r**ate(*abbr.* $QSHR$) to evaluate the selection accuracy

$$QSHR = \sum_{i=0}^{N} \frac{\bar{q}_i}{N}, \tag{14}$$

where N denotes the number of text streams, and $\bar{q}_i \in \{0, 1\}$. In the i-th text stream, $\bar{q}_i$=1 means the region annotated with "high" is hit, 0 otherwise.

Based on the selection mechanism, we further define the **r**ate of **c**orrectly **r**ecognizing selected text regions (*abbr.* RCR) to evaluate sequence-level recognition accuracy

$$RCR = \sum_{i=0}^{N} \frac{\bar{a}_i}{N}, \tag{15}$$

where $\bar{a}_i \in \{0, 1\}$. In the i-th text stream, $\bar{a}_i$=1 means the selected text region is correctly recognized, 0 otherwise.

[2]Available at https://davar-lab.github.io/opensource/dataset/lsvtd

D. End-to-end metrics

In previous methods, $MOTP$, $MOTA$ and ATA are generally used in the end-to-end evaluation, which evaluates performance in word-level recognition. That is, a predicted word is considered as a true positive if its IoU over ground truth is larger than 0.5 and the word recognition is correct. However, these metrics are not suited to our task because we only need to recognize the highest-quality text region in its scored text stream.

According to the selection-and-recognition strategy, we re-define new end-to-end metrics by considering two constraints: 1) The recognized results of selected regions should match the corresponding text transcriptions. 2) The temporal locations (frame ID) of selected regions should fall into the interval between the annotated starting and ending frame. In addition, the selected candidate should have a spatial overlap ratio (default by over 0.5) with its annotated bounding box. Thus we separately define the sequence-level recall REC_s and precision (PRE_s)

$$REC_s = \frac{N_r}{N_g}, \quad PRE_s = \frac{N_r}{N_d}, \tag{16}$$

for *constraint 1* and *constraint 2*, in which N_r, N_g and N_d separately denote the number of valid recalled streams, the number of total ground truth streams and the number of detected text streams. Correspondingly, the sequence-level F-score ($F\text{-}Score$) is denoted as

$$F\text{-}Score = \frac{2PRE_sREC_s}{PRE_s + REC_s} \tag{17}$$

by simultaneously considering PRE_s and REC_s.

VI. Experiments

A. Implementation Details

All of our work is built on the CAFFE framework with 32GB-Tesla-V100 GPUs. With the pre-trained TP [17] model on static images, we first train the spatial-temporal detector and text recommender separately, then both branches are jointly fine-tuned to obtain the end-to-end model.

Spatial-Temporal Detector. The architecture of the detection backbone is the same as TP [17], which is pre-trained on the 'Incidental Scene Text' dataset [15] and 'COCO-Text' dataset [94] by following [17], and then the model is fine-tuned on corresponding video training set such as IC13 or IC15. In the training stage, data augmentation used in [17] is employed, and *batch-size* is set to 4. We train the network by adopting 'SGD' with *learning rate*=$2 * 10^{-3}$, *momentum*=0.9 and *weight decay*=$5 * 10^{-4}$. Besides, text regions with a short side less than 10 pixels are ignored during training. While in the testing stage, we resize input images with the longer side 2000 and only conduct single-scale testing.

Text Recommender. The 'ResNet Backbone'+'Conv Blocks' backbone used in text recommender is adopted from the image encoder used in [41], and the shared 'BLSTM'+'ATT' module in quality scoring and recognition branch is an attention decoder used in [41], [43]. The network is pre-trained on the 8-million synthetic data [95] using 'Adadelta' by following [43], and further fine-tuned on corresponding datasets using SGD with the fixed learning rate of 10^{-4}. The loss weight λ_t in Equa. (10) and λ_1, λ_2, λ_3 in Equa. (12) are all empirically set to 1. In the text tracking part, the margin in triplet loss is set to 1.2, and MC is set to 1.08.

In the template generation network, the pre-trained recognition model follows the same architecture and training procedure as the recognition branch in text recommender. λ_d is set to 1 and *batch-size* is set to 9600. 'Adam' is used with *learning rate* = 0.0005 and *decay rate* = 0.94 for every 200 epochs.

Joint Training. Finally, we jointly fine-tune the whole network using the soft loss weight strategy mentioned in [17] for the other 40 epochs. 'SGD' optimization is adopted with *initial learning rate*=10^{-3}, *stepsize*=10 epochs and *gamma*=0.1.

B. Ablation Study

In this section, we separately evaluate the effects of spatial-temporal aggregation, text recommender and the whole framework. For a fair comparison, we here declare some baselines in our framework, i.e., (1) For detector, we treat the video text detection without the spatial-temporal strategy as our detection baseline (denoted by D-BASE), the spatial-temporal detection model as D-ST, and the end-to-end trained D-ST as D-ST (FREE). (2) For text recommender (*abbr*. TR), the experimental setting by training tracking, recognition and scoring branch separately is regarded as its baseline (denoted by TR-BASE), and we denote the end-to-end trained TR with three branches as TR (FREE).

1) Effects of spatial-temporal aggregation: We evaluate the proposed spatial-temporal module with three different detection backbones including EAST [16], Mask-RCNN [96] and TP [17]. As expected, the spatial-temporal mechanism can recall missing text regions caused by motion interferences, as shown in Table II. Compared to D-BASE, the spatial-temporal detection module D-ST can effectively improve REC by 4.02%/1.52%/4.27% and F-measure by 1.34%/0.64%/0.91%, but with PRE down by 4.31%/0.49%/3.72% on EAST, Mask-RCNN and TP, respectively. Actually, boosting recall performance is more important when facing very low recall results, which generally leads to precision decreasing. In addition, we find that the D-ST (FREE) can boost the F-measure.

We also verify the inference speed of three detection modules. To be fair, EAST (used in YORO), Mask-RCNN and TP share the same feature extraction backbone (ResNet50), and all images are resized to the longer side=2000. In Table II, EAST achieves the best detection speed, while TP has a moderate detection speed but with the best F-measure. However, the speed bottleneck of video text spotting usually lies in the following recognition module, as shown in the next Section.

2) Effects of text recommender: In IC13 and IC15, text regions are annotated as 3 quality levels ('low', 'moderate' and 'high'). Those streams containing at least two types of quality annotations are treated as our testing dataset. To evaluate the proposed text recommender, we compare our method with two commonly used recognition-and-voting strategies: (1) Using the predicted confidence (the average probability of generating

TABLE II
EFFECTS OF SPATIAL-TEMPORAL AGGREGATION ON IC13 WITH DIFFERENT BACKBONES. FPS MEANS FRAMES PER SECOND.

Backbone	Method	*REC*	*PRE*	*F-measure*	*FPS*
EAST [16]	D-BASE	56.21	**85.76**	67.91	**14.3**
	D-ST	**60.23**	81.45	**69.25**	
Mask-RCNN [96]	D-BASE	63.13	**71.71**	67.14	6.1
	D-ST	**64.65**	71.22	**67.78**	
TP [17]	D-BASE	65.53	**81.27**	72.56	8.8
	D-ST	**69.80**	77.55	73.47	
	D-ST(FREE)	68.37	79.70	**73.60**	

characters) of a word as the quality score (denoted by PCW). (2) Selecting the text region with the highest frequency of predicted results as the voted best one (denoted by HFP), which is similar to the *majority voting* strategy used in [3].

Performance evaluation. Table III shows the results. Compared to HFP, TR ($\mathcal{L}_{TR}$) in FREE significantly improves the *QSHR* performance by 7.73% on IC13 and 8.15% on IC15, and improves the *RCR* performance by 2.73% on IC13 and 3.31% on IC15. To demonstrate the effects of three learning branches in text recommender, we verify them as TR ($\mathcal{L}_S$), TR ($\mathcal{L}_S$+$\mathcal{L}_R$) and TR ($\mathcal{L}_S$+$\mathcal{L}_T$) in Table III. We find that either the joint training of tracking or recognition branch with scoring branch can boost the scoring results. The joint training of recognition, scoring and tracking branches can further boost the scoring performance.

TABLE III
EFFECTS OF TEXT RECOMMENDER ON IC13 AND IC15 COMPARED WITH OTHER FRAME SELECTION METHODS. TEMPLATE MEANS THE STANDARD TEMPLATE GENERATION STRATEGY. (.) REFERS TO THE LOSS APPLIED IN TR.

Method	Template Strategy	*QSHR* (IC13/IC15)	*RCR* (IC13/IC15)	*FPS*
PCW	-	74.55/75.83	66.06/66.32	4.52
HFP		75.32/76.34	68.30/68.56	
TR ($\mathcal{L}_S$)	K-Means (YORO[8])	77.89/79.69	68.89/69.41	324.58
TR ($\mathcal{L}_S$+$\mathcal{L}_T$)		78.64/80.36	69.12/69.82	
TR ($\mathcal{L}_S$+$\mathcal{L}_R$)		81.23/83.03	69.92/70.69	
TR ($\mathcal{L}_{TR}$)		81.74/83.29	70.18/70.95	
TR ($\mathcal{L}_S$)	Learning-based (FREE)	78.83/80.97	69.46/70.39	
TR ($\mathcal{L}_S$+$\mathcal{L}_T$)		80.19/81.51	69.84/70.79	
TR ($\mathcal{L}_S$+$\mathcal{L}_R$)		82.67/84.19	70.76/71.65	
TR ($\mathcal{L}_{TR}$)		**83.05/84.49**	**71.03/71.87**	

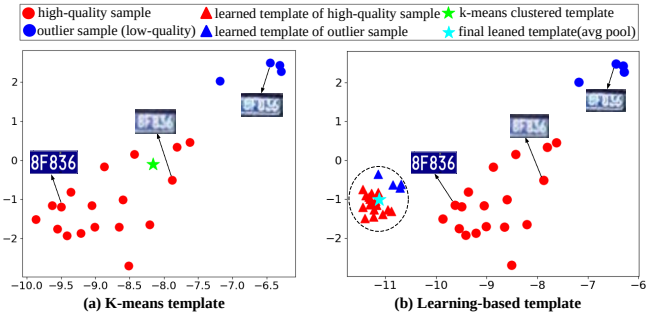


Fig. 7. Illustration of K-means based templates and the learning based templates.

In FREE, we design the new quality scoring branches instead of K-Means (used in YORO[8]), because K-Means is inherently sensitive to outlier samples and not robust to the complex scenarios. As shown in Figure 7.(a), outlier samples may cause the K-Means based template drifting away from high-quality samples. While the proposed learning-based strategy can alleviate the effect of outliers, keeping the learned template closer to high-quality samples (See Figure 7.(b)). Therefore, we design the learning-based template generation strategy, which is verified in Table III. The higher *QSHR* and *RCR* results compared to YORO [8] demonstrate the effectiveness of the enhanced template generation strategy.

For simplified representation, we use TR as TR ($\mathcal{L}_{TR}$) in the sequel.

Efficiency evaluation. Moreover, the text recommender only needs to recognize a text stream one-time, which can greatly decrease the computational cost. As shown in Table III, TR speeds up the recognition process averagely by 71 times compared to the frame-wise manner. In the end-to-end efficiency evaluation, the proposed selection-and-recognition framework is also much more efficient than the traditional one-by-one framework, as shown in Table IV. We find that the speed bottleneck of video text spotting usually lies in the recognition module, and the selection-and-recognition strategy can speed up the recognition process by 22.6 times. Equipped with different detection backbones (EAST, MASK-RCNN and TP), the proposed text recommender approach can largely speed up the end-to-end inference speed by 4.51, 2.96 and 3.58 times, respectively. Here we can see that text recommender can significantly improve the inference speed no matter what the detection backbone is. It is important to use the proposed text recommender in the end-to-end video text spotting, especially in the text-crowded cases (*e.g.* abundant text regions in each frame).

TABLE IV
EFFICIENCY COMPARISON BETWEEN SELECTING-AND-RECOGNITION (*abbr.* *Sa*R) AND ONE-BY-ONE (*abbr.* *Ob*O) STRATEGY ON IC13. MASK REFERS TO MASK-RCCNN. FPS MEANS FRAMES PER SECOND.

Modules	Backbone	FPS (SaR)	FPS (ObO)	Speedup (×)
Detection	EAST	14.30	14.30	1.0
	MASK	6.10	6.10	1.0
	TP	8.80	8.80	1.0
Tracking Recognition	ResNet[41]	33.88	33.95	1.0
		51.74	2.29	**22.6**
Total	EAST+ResNet[41]	8.42	1.87	**4.51**
	MASK+ResNet[41]	4.70	1.59	**2.96**
	TP+ResNet[41]	6.15	1.72	**3.58**

Extreme testing for text recommender. It is worth noticing that TR can still select the best one when handling text streams with a large proportion of low-quality text regions, while the voting strategy becomes useless. It implies that TR is more robust in complex and heavily distorted video scenarios. Therefore, we conduct extreme testing on a constituted *low-quality text stream set* by discarding all streams containing more than 40% highest quality text regions on IC13 and IC15. We calculate the *QSHR* and *RCR* on this set by checking

whether the highest quality of the text is hit and whether the selected text is correctly recognized. Table V gives the results and demonstrates that TR is more robust in complex and low-quality video scenarios.

TABLE V
EXTREME TESTING OF TR ON IC13 AND IC15 COMPARED WITH OTHER FRAME SELECTION METHODS.

Method	$QSHR$ (IC13/IC15)	RCR (IC13/IC15)
PCW	41.73/45.66	59.78/60.62
HFP	39.37/41.73	58.96/60.06
TR	**52.06/55.12**	**66.92/68.35**

3) Evaluation of the whole framework: To analyze the contributions of the above components, we verify the video text spotting on the popular IC15 dataset, as shown in Table VI. We find that

- Comparing to the detection baseline, the spatial-temporal strategy can steadily improve the video text spotting performance since it can recall more text under complex occasions.
- Compared to TR-BASE, TR can greatly improve the framework performance of PRE_s, REC_s and $F\text{-}score$ respectively, thanks to the complementarity of three branches for extracting discriminative features.
- The D-ST+TR pipeline achieves 4.85% F-score compared to D-BASE+TR-BASE. Moreover, the end-to-end optimization strategy can further enlarge the performance gain (5.18%), because the end-to-end training enables the utilization of recognition information for promoting the localization task precision (the 1.04% PRE_s gain compared with D-ST+TR).

These results demonstrate the effectiveness of text recommender as well as global optimization.

TABLE VI
THE END-TO-END EVALUATION ON IC15.

Modules	Pipeline strategies				End-to-end (FREE)
D-BASE	✓	✓			
D-ST			✓	✓	✓
TR-BASE	✓		✓		
TR		✓		✓	✓
PRE_s	71.85	73.06	68.31	70.48	71.52
REC_s	56.38	61.56	61.38	65.74	65.48
$F\text{-}score$	63.18	66.81	64.65	68.03	**68.36**

TABLE VII
DETECTION RESULTS ON IC13, IC15, YVT AND RT-1K, RESPECTIVELY.

Dataset	Method	REC	PRE	$F\text{-}measure$
IC13	Epshtein et al. [97]	32.53	39.80	35.94
	Khare et al. [11]	41.40	47.60	44.30
	Khare et al. [62]	55.90	57.91	51.70
	Zhao et al. [12]	47.02	46.30	46.65
	Shivakumara [13]	53.71	51.15	50.67
	Shivakumara [63]	57.00	61.00	59.00
	Yin et al. [9]	54.73	48.62	51.56
	Wu et al. [73]	68.00	63.00	65.00
	Wang et al. [10]	51.74	58.34	54.45
	Wang et al. [75]	58.67	71.90	62.65
	Yu et al. [77]	56.36	**82.36**	66.92
	YORO [8]	60.23	81.45	69.25
	D-BASE	65.53	81.27	72.56
	D-ST(FREE)	**68.37**	79.70	**73.60**
IC15	D-BASE	66.12	**82.58**	73.43
	D-ST(FREE)	**68.99**	80.77	**74.41**
YVT	Epshtein et al. [97]	76.00	68.00	72.00
	Zhao et al. [12]	41.00	34.00	37.00
	Mosleh et al. [60]	72.00	79.00	75.00
	Wu et al. [73]	73.00	81.00	77.00
	Shivakumara[63]	73.00	79.00	76.00
	Yu et al. [77]	71.03	89.12	79.05
	D-BASE	78.21	**91.93**	84.51
	D-ST(FREE)	**81.66**	90.30	**85.76**
RT-1K	Reddy et al. [91]	41.00	44.00	42.00
	D-BASE	41.39	**64.83**	50.52
	D-ST(FREE)	**43.39**	63.02	**51.39**

TABLE VIII
TRACKING PERFORMANCE EVALUATION ON IC13 AND IC15, YVT AND RT-1K, RESPECTIVELY. THE SUFFIX 'D' MEANS TRACKING IS APPLIED FOR DETECTION.

Dataset	Methods	ATA_D	$MOTP_D$	$MOTA_D$
IC13	IC13's base [89]	0.00	0.63	-0.09
	TextSpotter [98]	0.12	0.67	0.27
	Nguyen et al. [2]	0.15	-	-
	YORO [8]	0.64	0.75	0.67
	T-BASE	0.59	0.76	0.68
	TR(FREE)	**0.67**	**0.76**	**0.71**
IC15	Stradvision-1 [15]	0.32	0.71	0.48
	Deep2Text-I [15]	0.45	0.71	0.41
	Wang et al. [3]	0.56	0.70	0.57
	Yang et al. [74]	0.61	**0.79**	0.66
	YORO [8]	0.65	0.76	0.68
	T-BASE	0.61	0.77	0.68
	TR(FREE)	**0.68**	0.77	**0.72**
YVT	DR+T [2]	0.31	-	-
	T-BASE	0.46	0.78	0.52
	TR(FREE)	**0.51**	**0.78**	**0.54**
RT-1K	Reddy et al. [91]	0.01	0.07	-0.11
	T-BASE	0.12	0.71	0.01
	TR(FREE)	**0.17**	**0.71**	**0.03**

C. Comparison with State-of-The-Arts

In this section, we evaluate our method and compare it with previous methods on several benchmarks including IC13, IC15, YVT and RT-1K.

1) Comparison on detection: **Evaluation on IC13 and YVT**. From Table VII, we find that the D-BASE already outperforms the existing approaches by a large margin thanking to the robust TP, but also suffers from low recall due to the complicated motion scenarios. As demonstrated in the Ablation Section, D-ST(FREE) can improve *REC* by 2.84%/3.45% and *F-measure* by 1.04%/1.25% on both IC13 and YVT. Compared to state-of-the-art methods such as YORO [8] and Yu et al. [77] on IC13 and YVT, FREE obtains 4.35% and 6.71% *F-measure* gains significantly.

Evaluation on IC15 and RT-1K. For IC15, there are no results reported. We only list the testing results. For the recent released RT-1K, we compare our results with the best-reported performance in the RT-1K paper [91], which demonstrates the effectiveness of our method.

TABLE IX
THE TRADITIONAL END-TO-END EVALUATION ON IC15, YVT AND RT-1K, RESPECTIVELY. THE SUFFIX 'R' MEANS TRACKING IS APPLIED FOR MEASURING RECOGNITION.

Dataset	Method	$MOTP_R$	$MOTA_R$	ATA_R
IC15	Stradvision [15]	0.69	0.57	0.29
	Deep2Text [15]	0.62	0.35	0.19
	Wang et al. [3]	0.70	0.69	0.60
	YORO [8]	0.76	0.69	0.63
	Ours	**0.78**	**0.72**	**0.65**
YVT	DR+T [2]	-	-	-
	Ours	0.81	0.44	0.41
RT-1K	Reddy et al. [91]	-	-	-
	Ours	0.72	0.03	0.12

2) Comparison on tracking: Table VIII shows the comparing results on IC13 and IC15.

Evaluation on IC13. T-BASE (D-BASE+TR ($\mathcal{L}_T$)) outperforms the reported results by a large margin 0.44 on ATA_D, 0.09 on $MOTP_D$ and 0.41 on $MOTA_D$. TR(FREE) (D-ST+TR) can further separately improve the performance by 0.08 and 0.03 on ATA_D and $MOTA_D$, and maintain the $MOTP_D$ performance, attributed to the spatial-temporal detector and the text recommender.

Evaluation on IC15. T-BASE also achieves comparable results with previous methods. Comparing to the best-reported result [74], TR(FREE) significantly improves the ATA_D and $MOTA_D$ by 0.07 and 0.06, but falls behind on $MOTP_D$. However, [74] points out that ATA_D is the most important metric because ATA_D measures the tracking performance over all the text.

Evaluation on YVT and RT-1K. There are few reported tracking results on YVT and RT-1K. We also list the predicting results of our method, which demonstrates its robustness.

3) Comparison on classic end-to-end evaluation: Conventionally, we first place the frame-wise recognition results on IC15 by referring to the previous works. To be fair, we use the recognition output of the selected text region as the predicting result of each text region in a text stream. It means that each text region in a text stream shares the same predicting result. In this way, we can calculate the frame-wise performance of FREE and compare it with previous methods. Table IX shows that our method also achieves the remarkable state-of-the-art under this experimental setting. There are no reported end-to-end evaluation results on YVT and RT-1K. Here, we also list the corresponding results on YVT and RT-1K.

D. Challenges on IC13, IC15, YVT, RT-1K and LSVTD

We here list all testing results on IC13, IC15, YVT, RT-1K and LSVTD, as shown in Table X.

1) Overall comparisons: Compared to the state-of-the-art [8], [10], [2], [98], [74], [73], [77], [91] on all datasets, the new framework FREE achieves better performance on detection, tracking and end-to-end evaluation. Besides, results on LSVTD and RD-1K are largely lower than those on IC13, IC15 and YVT, which reflects that LSVTD and RD-1K are more challenging.

2) Challenges on 21 scenes of LSVTD: We further detail the performance of individual scenarios from LSVTD, and the challenge of spotting video text is ranked based on the F-score performance, as shown in Figure 8. We can see that the difficulty of spotting video text is very different with F-score ranging from 4.3% to 72.1%. For example, compared to indoor cases (books opening, office building), spotting video text outdoor such as *city road* and *harbor surveillance* is more challenging largely due to extremely complex background and fierce motion.

3) Evaluating LSVTD with more popular models: We here evaluate more popular methods on LSVTD. All evaluations are conducted with the selecting-and-recognition strategy. Concretely, we select EAST, Mask-RCNN and TP as the detection module, in which ResNet50 is used for feature extraction. For text recommender module, the feature backbone is the same as that used in [41]. Then we select two popular sequence decoding structures for the quality scoring branch and recognition branch: the attention-based decoder ('BLSTM'+'ATT') [41], [43] and the CTC-based decoder ('BLSTM'+'CTC') [38]. Thus, there are six kinds of module combinations for evaluating the released LSVTD, as shown in Table X. Note that, 'F-measure', 'ATA_D' and 'F-score' are comprehensive evaluation metrics for evaluating the detection, tracking and end-to-end performance.

We see that EAST obtains the worst results on detection stage, while the TP model performs better than Mask-RCNN and EAST. Integrated with three detectors, the attention-based recognition model always outperforms CRNN on both tracking and end-to-end evaluation. Because attention-based decoder can handle more complicated text regions. Therefore, 'TP+ATT' is the best combination among the pipeline frameworks. Furthermore, compared to 'TP+ATT', FREE can further boost the performance of spotting video text.

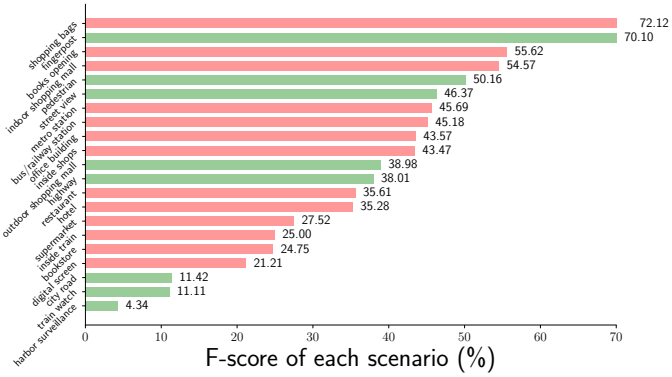


Fig. 8. Illustration of *F-score* performance on 21 scenarios. Indoor/Outdoor scenarios are highlighted in red/green.

E. Challenges on ICDAR 'Text in Video' Competition

We also evaluate our method on the *'Text Localization'* and *'End-to-End'* benchmarks of *'Text in Videos'* on ICDAR official website[3], the evaluation metrics include MOTA, MOTP and IDF1 (details can be referred on the website). We achieve

[3]http://rrc.cvc.uab.es/?ch=3&com=evaluation&task=1

TABLE X
THE OVERALL PERFORMANCE OF FREE ON IC13, IC15, YVT, RT-1K AND LSVTD. MASK, ATT AND CRNN SEPARATELY MEAN THE MASK-RCNN, ATTENTION-BASED SEQUENCE DECODER AND CTC-BASED DECODER. YORO FRAMEWORK IS EQUIVALENT TO THE EAST+ATT FRAMEWORK, AND THE SUPERSCRIPT * MEANS THE METHOD IS EVALUATED ON THE REFINED LSVTD.

Dataset	Methods	Detection			Tracking			End–to-end		
		REC	PRE	$F\text{-}measure$	ATA_D	$MOTP_D$	$MOTA_D$	PRE_s	REC_s	$F\text{-}score$
IC13	Best of IC13	51.74 [10]	58.34 [10]	54.45 [10]	0.15 [2]	0.67 [98]	0.27 [98]	-	-	-
	YORO [8]	60.23	**81.45**	69.25	0.64	0.75	0.67	67.03	**64.78**	65.89
	FREE	**68.37**	79.70	**73.60**	**0.66**	**0.76**	**0.71**	**70.23**	63.06	**66.45**
IC15	Best of IC15	-	-	-	0.61 [74]	0.79 [74]	0.66 [74]	-	-	-
	YORO [8]	64.02	81.26	71.62	0.65	0.76	0.68	68.28	**67.21**	67.74
	FREE	**69.83**	**84.39**	**76.42**	**0.68**	**0.77**	**0.72**	**71.52**	65.48	**68.36**
YVT	Best of YVT	73.00[73]	89.12[77]	79.05[77]	0.31[2]	-	-	-	-	-
	FREE	**81.66**	**90.30**	**85.76**	**0.51**	0.78	0.54	48.94	56.10	52.27
RT-1K	Best of RT-1K	41.00[91]	44.00[91]	42.00[91]	0.01[91]	0.07[91]	-0.11[91]	-	-	-
	FREE	**43.39**	**63.02**	**51.39**	**0.17**	**0.72**	**0.03**	34.82	20.74	26.00
LSVTD	EAST+CRNN	43.81	54.00	48.37	34.70	68.67	36.63	40.16	28.11	33.07
	YORO* [8]				34.97	68.45	37.09	41.01	30.78	35.17
	MASK+CRNN	**70.98**	54.48	61.64	37.38	73.8	37.4	53.90	30.69	39.11
	MASK+ATT				39.36	**74.67**	41.49	**56.47**	30.05	39.23
	TP+CRNN	64.61	**60.65**	62.57	43.01	73.05	42.81	53.08	34.36	41.72
	TP+ATT				43.33	72.91	43.62	53.79	**35.20**	42.55
	FREE*	70.30	58.70	**63.98**	**44.26**	72.54	**45.19**	55.91	34.97	**43.03**

the remarkable state-of-the-art on both benchmarks, surpassing previous methods to a large extent. For example in the *'End-to-End'* evaluation, we achieve over 23% MOTA gain compared with the 2-nd approach.

VII. CONCLUSION

In this paper, we propose a fast and robust video text spotting framework named as FREE by integrating a well-designed spatial-temporal video text detector and a novel text recommender in an end-to-end trainable manner. The video text detector is responsible for recalling more text by referring to the relation between different frames. The text recommender is designed for selecting the highest-quality text from tracked text streams and then only recognizing the selected text region once, which not only ignores the inferences of low-quality text, and also significantly speeds up the recognition process. Besides, the end-to-end trainable mechanism further improves the video text spotting performance. Finally, we release a larger-scale video scene text dataset for better evaluating video text spotting algorithms. In the future, we'll further improve the efficiency of video text detector.

ACKNOWLEDGMENT

Fei Wu was supported in part by NSFC under Grant 61625107. Shuigeng Zhou was partially supported by National Natural Science Foundation of China (NSFC) under grant No. 61972100, 2019 Special Fund for Artificial Intelligence Innovation & Development, Shanghai Economy and Information Technology Commission (SHEITC), and the Science and Technology Commission of Shanghai Municipality Project under Grant 19511120700.

REFERENCES

[1] C. Merino-Gracia and M. Mirmehdi, "Real-time text tracking in natural scenes," *IET Computer Vision*, vol. 8, no. 6, pp. 670–681, 2014.

[2] P. X. Nguyen, K. Wang, and S. Belongie, "Video text detection and recognition: Dataset and benchmark," in *WACV*, 2014, pp. 776–783.

[3] X. Wang, Y. Jiang, S. Yang, X. Zhu, W. Li, P. Fu, H. Wang, and Z. Luo, "End-to-End Scene Text Recognition in Videos Based on Multi Frame Tracking," in *ICDAR*, vol. 1, 2017, pp. 1255–1260.

[4] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An End-to-End TextSpotter With Explicit Alignment and Attention," in *CVPR*, 2018, pp. 5020–5029.

[5] H. Li, P. Wang, and C. Shen, "Towards End-To-End Text Spotting With Convolutional Recurrent Neural Networks," in *ICCV*, 2017, pp. 5238–5246.

[6] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast Oriented Text Spotting with a Unified Network," in *CVPR*, 2018, pp. 5676–5685.

[7] 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 *ECCV*, 2018, pp. 71–88.

[8] Z. Cheng, J. Lu, Y. Niu, S. Pu, F. Wu, and S. Zhou, "You Only Recognize Once: Towards Fast Video Text Spotting," in *ACM MM*. ACM, 2019, pp. 855–863.

[9] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE TPAMI*, vol. 36, no. 5, pp. 970–983, 2014.

[10] L. Wang, Y. Wang, S. Shan, and F. Su, "Scene Text Detection and Tracking in Video with Background Cues," in *ICMR*, 2018, pp. 160–168.

[11] V. Khare, P. Shivakumara, and P. Raveendran, "A new Histogram Oriented Moments descriptor for multi-oriented moving text detection in video," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7627–7640, 2015.

[12] X. Zhao, K.-H. Lin, Y. Fu, Y. Hu, Y. Liu, and T. S. Huang, "Text from corners: a novel approach to detect text and caption in videos," *IEEE TIP*, vol. 20, no. 3, pp. 790–799, 2011.

[13] P. Shivakumara, R. P. Sreedhar, T. Q. Phan, S. Lu, and C. L. Tan, "Multioriented video scene text detection through bayesian classification and boundary growing," *IEEE TCSVT*, vol. 22, no. 8, pp. 1227–1235, 2012.

[14] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *ICCV*. IEEE, 2011, pp. 1457–1464.

[15] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "ICDAR 2015 competition on robust reading," in *ICDAR*, 2015, pp. 1156–1160.

[16] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An Efficient and Accurate Scene Text Detector," in *CVPR*, 2017, pp. 5551–5560.

[17] L. Qiao, S. Tang, Z. Cheng, Y. Xu, Y. Niu, S. Pu, and F. Wu, "Text Perceptron: Towards End-to-End Arbitrary-Shaped Text Spotting," in *AAAI*, 2020, pp. 11 899–11 907.

[18] Y. Zhuang, M. Cai, X. Li, X. Luo, Q. Yang, and F. Wu, "The Next Breakthroughs of Artificial Intelligence: The Interdisciplinary Nature of AI," *Engineering*, vol. 6, no. 3, pp. 245–247, 2020.

[19] N. Lei, D. An, Y. Guo, K. Su, S. Liu, Z. Luo, S.-T. Yau, and X. Gu, "A Geometric Understanding of Deep Learning," *Engineering*, vol. 6, no. 3, pp. 361–374, 2020.

[20] Y.-t. Zhuang, F. Wu, C. Chen, and Y.-h. Pan, "Challenges and opportunities: from big data to knowledge in AI 2.0," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 1, pp. 3–14, 2017.

[21] Y. Zhu, T. Gao, L. Fan, S. Huang, M. Edmonds, H. Liu, F. Gao, C. Zhang, S. Qi, Y. N. Wu *et al.*, "Dark, Beyond Deep: A Paradigm Shift to Cognitive AI with Humanlike Common Sense," *Engineering*, vol. 6, no. 3, pp. 310–345, 2020.

[22] K. Kuang, L. Li, Z. Geng, L. Xu, K. Zhang, B. Liao, H. Huang, P. Ding, W. Miao, and Z. Jiang, "Causal Inference," *Engineering*, vol. 6, pp. 253–263, 2020.

[23] L.-k. Zhou, S.-l. Tang, J. Xiao, F. Wu, and Y.-t. Zhuang, "Disambiguating named entities with deep supervised learning via crowd labels," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 1, pp. 97–106, 2017.

[24] X.-y. Duan, S.-l. Tang, S.-y. Zhang, Y. Zhang, Z. Zhao, J.-r. Xue, Y.-t. Zhuang, and F. Wu, "Temporality-enhanced knowledge memory network for factoid question answering," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 104–115, 2018.

[25] Y. Xu, C. Zhang, Z. Cheng, J. Xie, Y. Niu, S. Pu, and F. Wu, "Segregated Temporal Assembly Recurrent Networks for Weakly Supervised Multiple Action Detection," in *AAAI*, vol. 33, 2019, pp. 9070–9078.

[26] C. Zhang, Y. Xu, Z. Cheng, Y. Niu, S. Pu, F. Wu, and F. Zou, "Adversarial Seeded Sequence Growing for Weakly-Supervised Temporal Action Localization," in *ACM MM*, 2019, pp. 738–746.

[27] Y. Pan, "Multiple Knowledge Representation of Artificial Intelligence," *Engineering*, vol. 6, pp. 216–217, 2019.

[28] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "WordSup: Exploiting Word Annotations for Character Based Text Detection," in *ICCV*, 2017, pp. 4940–4949.

[29] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection," *arXiv preprint arXiv:1706.09579*, 2017.

[30] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai, "Rotation-Sensitive Regression for Oriented Scene Text Detection," in *CVPR*, 2018, pp. 5909–5918.

[31] Y. Liu and L. Jin, "Deep Matching Prior Network: Toward Tighter Multi-oriented Text Detection," in *CVPR*, 2017, pp. 3454–3461.

[32] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-Oriented Scene Text Detection via Rotation Proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.

[33] B. Shi, X. Bai, and S. Belongie, "Detecting Oriented Text in Natural Images by Linking Segments," in *CVPR*, 2017, pp. 2550–2558.

[34] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep Direct Regression for Multi-Oriented Scene Text Detection," in *ICCV*, 2017, pp. 745–753.

[35] F. Wang, L. Zhao, X. Li, X. Wang, and D. Tao, "Geometry-Aware Scene Text Detection With Instance Transformation Network," in *CVPR*, 2018, pp. 1381–1389.

[36] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *ICML*, 2006, pp. 369–376.

[37] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large Scale System for Text Detection and Recognition in Images," in *SIGKDD*, 2018, pp. 71–79.

[38] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE TPAMI*, vol. 39, no. 11, pp. 2298–2304, 2017.

[39] J. Wang and X. Hu, "Gated recurrent convolution neural network for OCR," in *NIPS*, 2017, pp. 335–344.

[40] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, "Edit Probability for Scene Text Recognition," in *CVPR*, 2018, pp. 1508–1516.

[41] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *ICCV*, 2017, pp. 5086–5094.

[42] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "AON: Towards Arbitrarily-Oriented Text Recognition," in *CVPR*, 2018, pp. 5571–5579.

[43] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust Scene Text Recognition with Automatic Rectification," in *CVPR*, 2016, pp. 4168–4176.

[44] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: An attentional scene text recognizer with flexible rectification," *IEEE TPAMI*, pp. 1–1, 2018.

[45] W. Feng, W. He, F. Yin, X. Zhang, and C. Liu, "Textdragon: An end-to-end framework for arbitrary shaped text spotting," in *ICCV*, 2019, pp. 9076–9085.

[46] 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 *CVPR*, 2020, pp. 9809–9818.

[47] S. Qin, A. Bissaco, M. Raptis, Y. Fujii, and Y. Xiao, "Towards Unconstrained End-to-End Text Spotting," in *ICCV*, 2019, pp. 4704–4714.

[48] Y. Sun, C. Zhang, Z. Huang, J. Liu, J. Han, and E. Ding, "TextNet: Irregular Text Reading from Images with an End-to-End Trainable Network," in *ACCV*, 2018.

[49] H. Wang, P. Lu, H. Zhang, M. Yang, X. Bai, Y. Xu, M. He, Y. Wang, and W. Liu, "All You Need Is Boundary: Toward Arbitrary-Shaped Text Spotting," in *AAAI*. AAAI Press, 2020, pp. 12 160–12 167.

[50] L. Xing, Z. Tian, W. Huang, and M. R. Scott, "Convolutional Character Networks," in *ICCV*, 2019, pp. 9126–9136.

[51] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: a comprehensive survey," *IEEE TIP*, vol. 25, no. 6, pp. 2752–2773, 2016.

[52] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE TPAMI*, vol. 25, no. 12, pp. 1631–1639, 2003.

[53] H. Goto and M. Tanaka, "Text-tracking wearable camera system for the blind," in *ICDAR*, 2009, pp. 141–145.

[54] M. Tanaka and H. Goto, "Autonomous text capturing robot using improved DCT feature and text tracking," in *ICDAR*, vol. 2, 2007, pp. 1178–1182.

[55] H. Shiratori, H. Goto, and H. Kobayashi, "An efficient text capture method for moving robots using dct feature and text tracking," in *ICPR*, vol. 2, 2006, pp. 1050–1053.

[56] M. Tanaka and H. Goto, "Text-tracking wearable camera system for visually-impaired people," in *ICPR*, 2008, pp. 1–4.

[57] L. Gómez and D. Karatzas, "MSER-based real-time text detection and tracking," in *ICPR*, 2014, pp. 3110–3115.

[58] R. Minetto, N. Thome, M. Cord, N. J. Leite, and J. Stolfi, "Snoopertrack: Text detection and tracking for outdoor videos," in *ICIP*, 2011, pp. 505–508.

[59] Z.-Y. Zuo, S. Tian, W.-y. Pei, and X.-C. Yin, "Multi-strategy tracking based text detection in scene videos," in *ICDAR*, 2015, pp. 66–70.

[60] A. Mosleh, N. Bouguila, and A. B. Hamza, "Automatic Inpainting Scheme for Video Text Detection and Removal," *IEEE TIP*, vol. 22, no. 11, pp. 4460–4472, 2013.

[61] C. Yang, X.-C. Yin, W.-Y. Pei, S. Tian, Z.-Y. Zuo, C. Zhu, and J. Yan, "Tracking based multi-orientation scene text detection: A unified framework with dynamic programming," *IEEE TIP*, vol. 26, no. 7, pp. 3235–3248, 2017.

[62] V. Khare, P. Shivakumara, R. Paramesran, and M. Blumenstein, "Arbitrarily-oriented multi-lingual text detection in video," *Multimedia Tools and Applications*, vol. 76, no. 15, pp. 16 625–16 655, 2017.

[63] P. Shivakumara, L. Wu, T. Lu, C. L. Tan, M. Blumenstein, and B. S. Anami, "Fractals based multi-oriented text detection system for recognition in mobile video images," *Pattern Recognition*, vol. 68, pp. 158–174, 2017.

[64] L. Wang, J. Shi, Y. Wang, and F. Su, "Video Text Detection by Attentive Spatiotemporal Fusion of Deep Convolutional Features," in *ACM MM*. ACM, 2019, pp. 66–74.

[65] Y. Wang, L. Wang, F. Su, and J. Shi, "Video Text Detection with Fully Convolutional Network and Tracking," in *ICME*. IEEE, 2019, pp. 1738–1743.

[66] V. Fragoso, S. Gauglitz, S. Zamora, J. Kleban, and M. Turk, "TranslatAR: A mobile augmented reality translator," in *WACV*, 2011, pp. 497–502.

[67] Y. Na and D. Wen, "An effective video text tracking algorithm based on sift feature and geometric constraint," in *PRCM*, 2010, pp. 392–403.

[68] M. Petter, V. Fragoso, M. Turk, and C. Baur, "Automatic text detection for mobile augmented reality translation," in *Workshop on ICCV*, 2011, pp. 48–55.

[69] X. Rong, C. Yi, X. Yang, and Y. Tian, "Scene text recognition in multiple frames based on text tracking," in *ICME*, 2014, pp. 1–6.

[70] S. Tian, W.-Y. Pei, Z.-Y. Zuo, and X.-C. Yin, "Scene Text Detection in Video by Learning Locally and Globally," in *IJCAI*, 2016, pp. 2647–2653.

[71] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *ECCV*, 2014, pp. 127–141.

[72] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
[73] L. Wu, P. Shivakumara, T. Lu, and C. L. Tan, "A New Technique for Multi-Oriented Scene Text Line Detection and Tracking in Video," *IEEE TMM*, vol. 17, no. 8, pp. 1137–1152, 2015.
[74] X.-H. Yang, W. He, F. Yin, and C.-L. Liu, "A Unified Video Text Detection Method with Network Flow," in *ICDAR*, vol. 1, 2017, pp. 331–336.
[75] Y. Wang, L. Wang, and F. Su, "A robust approach for scene text detection and tracking in video," in *PCM*. Springer, 2018, pp. 303–314.
[76] X. Wang, X. Feng, and Z. Xia, "Scene video text tracking based on hybrid deep text detection and layout constraint," *Neurocomputing*, vol. 363, pp. 223–235, 2019.
[77] H. Yu, C. Zhang, X. Li, J. Han, E. Ding, and L. Wang, "An End-to-end Video Text Detector with Online Tracking," in *ICDAR*. IEEE, 2019, pp. 601–606.
[78] J. Greenhalgh and M. Mirmehdi, "Recognizing Text-Based Traffic Signs," *IEEE TITS*, vol. 16, no. 3, pp. 1360–1369, 2015.
[79] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *AS-SPCC*, 2000, pp. 153–158.
[80] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-Guided Feature Aggregation for Video Object Detection," in *ICCV*, 2017, pp. 408–417.
[81] D. Chen, H. Li, T. Xiao, S. Yi, and X. Wang, "Video Person Re-Identification With Competitive Snippet-Similarity Aggregation and Co-Attentive Snippet Embedding," in *CVPR*, 2018, pp. 1169–1178.
[82] J. Si, H. Zhang, C.-G. Li, J. Kuen, X. Kong, A. C. Kot, and G. Wang, "Dual Attention Matching Network for Context-Aware Feature Sequence based Person Re-Identification," in *CVPR*, 2018, pp. 5363–5372.
[83] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," in *3DV*, 2016, pp. 565–571.
[84] M. Lin, Q. Chen, and S. Yan, "Video Person Re-Identification With Competitive Snippet-Similarity Aggregation and Co-Attentive Snippet Embedding," in *CVPR*, 2018, pp. 1169–1178.
[85] Y. Guo and N. Cheung, "Efficient and Deep Person Re-identification Using Multi-level Similarity," in *CVPR*, 2018, pp. 2335–2344.
[86] M. Lin, Q. Chen, and S. Yan, "Network In Network," in *ICLR*, 2013.
[87] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823.
[88] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, 2006, pp. 1735–1742.
[89] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "ICDAR 2013 robust reading competition," in *ICDAR*, 2013, pp. 1484–1493.
[90] C. Merino and M. Mirmehdi, "A framework towards realtime detection and tracking of text," in *CBDAR*, 2007, pp. 10–17.
[91] S. Reddy, M. Mathew, L. Gomez, M. Rusinol, D. Karatzas., and C. V. Jawahar, "RoadText-1K: Text Detection & Recognition Dataset for Driving Videos," in *ICRA*, 2020.
[92] Z. Zhong, L. Sun, and Q. Huo, "An Anchor-Free Region Proposal Network for Faster R-CNN based Text Detection Approaches," *IJDAR*, vol. 22, no. 3, pp. 315–327, 2019.
[93] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol," *IEEE TPAMI*, vol. 31, no. 2, pp. 319–336, 2008.
[94] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images," *arXiv preprint arXiv:1601.07140*, 2016.
[95] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition," *arXiv preprint arXiv:1406.2227*, 2014.
[96] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017, pp. 2980–2988.
[97] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *CVPR*, 2010, pp. 2963–2970.
[98] L. Neumann and J. Matas, "On combining multiple segmentations in scene text recognition," in *ICDAR*, 2013, pp. 523–527.

Zhanzhan Cheng received his B.S. degree from the School of Electronic & Information Engineering, Xian Jiaotong University, Xian, China in 2013. He received his M.S. degree from the School of Computer Science, Fudan University, Shanghai, China in 2016. He is currently a Ph.D. candidate with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. He is also currently with Hikvision Research Institute, Hangzhou, China. His research has focused on computer vision and machine learning, including scene text spotting, video analysis and intelligent systems.

Jing Lu received his B.S. degree from the School of Electronic Engineering and Information Science, the University of Science and Technology of China (USTC) in 2014. And graduated Shanghai Institute of Microsystem and Information Technology of Chinese Academy of Science in 2017. His main research interests include video text detection, recognition and image quality assessment.

Baorui Zou received his B.S. degree from the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China in 2018. He is currently a M.S. candidate with the School of Computer Science, Fudan University, Shanghai, China. His research has focused on scene text spotting and quality assessment.

Liang Qiao received his B.S and M.S degrees from the School of Software, Shanghai Jiao Tong University (SJTU), China in 2015 and 2018. He is currently an algorithm engineer in Hikvision Insititue. His main research interests include scene text detection and End-to-End text spotting.

Yunlu Xu received her B.S. and M.S degree from the School of Cyberspace Security Engineering, Shanghai Jiao Tong University, Shanghai, China in 2015 and 2018. She is a researcher in Hikvision Research Institute, Hangzhou, China. Her research has focused on computer vision and machine learning, including scene text recognition, video analysis.

Shiliang Pu received the PhD degree in applied optics from the University of Rouen, Mont-Saint-Aignan, France, in 2005. He is currently the executive vice director of the Research Institute with Hikvision, Hangzhou, China. He is also responsible for the company's technology research and development work on video intelligent analysis, image processing, coding, and decoding. His current research interests include image processing and pattern recognition.

Yi Niu received his B.S. degree from the department of automation, Huazhong University of Science and Technology, Wuhan, China in 2002. He received his M.S. degree from the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, China in 2007. He is also currently with Hikvision Research Institute, Hangzhou, China. His research has focused on intelligent transportation system.

Fei Wu received the B.Sc. degree in computer science from Lanzhou University, Lanzhou, China, in 1996, the M.Sc. degree in computer science from the University of Macau, Macau, China, in 1999, and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2002. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University. He was a Visiting Scholar with Prof. B. Yu's Group, University of California at Berkeley, Berkeley, CA, USA, from 2009 to 2010. His current research interests include multimedia retrieval, sparse representation, and machine learning.

Shuigeng Zhou is a professor of School of Computer Science, Fudan University, Shanghai, China. He received his Bachelor degree from Huazhong University of Science and Technology (HUST) in 1988, his Master degree from University of Electronic Science and Technology of China (UESTC) in 1991, and his PhD of Computer Science from Fudan University in 2000. He served in Shanghai Academy of Spaceflight Technology from 1991 to 1997, as an engineer and a senior engineer (since August 1995) respectively. He was a post-doctoral researcher in State Key Lab of Software Engineering, Wuhan University from 2000 to 2002. His research interests include big data management and analysis, artificial intelligence, and bioinformatics. He has published more than 200 papers in domestic and international journals (including ACM TITS, IEEE TKDE, IEEE TPDS, VLDB Journal, IEEE TCBB, Bioinformatics etc.) and conferences (including SIGMOD, VLDB, ICDE, SIGKDD, SIGIR, AAAI, IJCAI, ICCV, CVPR, SODA, ISMB and RECOMB etc.). Currently he is a senior member of IEEE and a member of ACM.