2019 IEEE International Conference on Multimedia and Expo (ICME)

VIDEO TEXT DETECTION WITH FULLY CONVOLUTIONAL NETWORK AND TRACKING

Yang Wang, Lan Wang, Feng Su[*†] *and Jiahao Shi*

State Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
suf@nju.edu.cn

ABSTRACT

Scene text in videos carries rich semantic information that is of great value in various content-based video applications. In this paper, we propose an effective fully convolutional network model for detecting text in videos based on a novel refine block structure. The model hierarchically exploits low-level features from earlier convolutions to refine high-level semantic features, thereby fusing multi-resolution features extracted from the frame to generate high-resolution semantic feature maps for better capturing widely varied appearances of video text. We further complement the individual-frame detection with an efficient correlation filter based text tracking mechanism, and enhance the overall detection performance by matching and combining detection and tracking results. Experiments on public scene text video datasets demonstrate the state-of-the-art performance of the proposed method.

Index Terms— Scene text, video, detection, tracking, fully convolutional network

1. INTRODUCTION

Scene text in the videos usually carries rich semantic information about the video content, and often plays an important role in various video applications such as video analysis, classification, indexing and retrieval. Compared to text appearing in natural scene images, scene text in the videos has some common attributes such as varied appearances (e.g. size, orientation, color, style) and complex context (e.g. background, lighting condition), while it also has specific characteristics such as temporal correlations on appearance and position between adjacent frames as well as degradations resulting from motions, which bring extra difficulties to reliable detection of text in the videos.

To detect scene text in one video sequence, one can exploit the existing large numbers of text detection methods designed for static images, which can be roughly divided into three groups: 1) connected component based methods [1, 2] that exploit connected component analysis to extract character candidates; 2) region based methods [3] that shift a multi-scale window on the image and discriminate the window region as text candidate or not with some classifiers; 3) deep neural network (DNN) based methods [4, 5, 6] that are built on variant DNN models such as convolutional neural network (CNN) and recurrent neural network (RNN) and generally can be further divided into two categories: proposals based methods [4, 5] that filter proposals generated by some object detection frameworks, and direct regression methods [6] that exploit Fully Convolutional Network (FCN) to predict the score map and coordinates of the text directly.

To further exploit temporal correlations of text cues across video frames to assist or rectify the detection, some multi-frame based video text detection methods [7, 8, 9] have also been proposed on the basis of text detection techniques for static images, in which, variant techniques such as spatial-temporal analysis, multi-frame integration and tracking are exploited to improve the holistic detection performance. However, due to the diversity of scene text's appearance and quality, to reliably localize scene text in complex video context remains a highly challenging task.

In this paper, we propose a robust scene text detection method for videos that combines a novel fully convolutional network model and effective text tracking mechanisms. The key contributions of our method are summarized as follows:

- We propose a novel *refine block* structure and on the basis of it, we construct an effective fully convolutional network model for detecting scene text in video frames. The model hierarchically exploits low-level features from earlier convolutions to refine high-level semantic features, thereby generating high-resolution semantic feature maps for better capturing widely varied appearances of video text.

- We further exploit a robust correlation filter based tracking algorithm for video text, and combine detection and tracking results to filter out false detections and recover missed text, which enhances holistic detection performance on the basis of individual-frame detection.

*Corresponding author

†Research supported by the Natural Science Foundation of Jiangsu Province of China under Grant No. BK20171345 and the National Natural Science Foundation of China under Grant Nos. 61003113, 61321491, 61672273.

978-1-5386-9552-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ICME.2019.00299

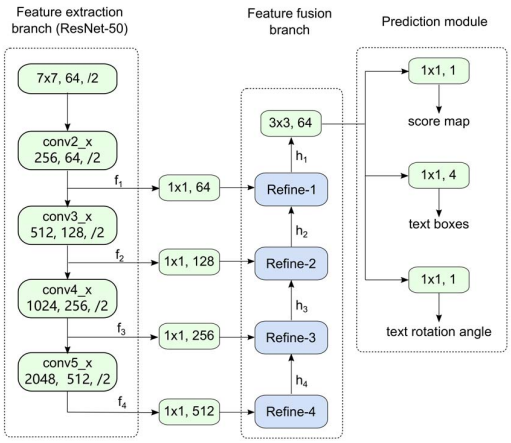

Fig. 1. The architecture of the proposed fully convolutional network model for text detection.

- The proposed method achieves better text detection results than state-of-the-art methods on public scene text video datasets.

2. TEXT DETECTION

Both low-level location-related information and high-level semantic information play important roles in detecting text from images. Meanwhile, high-resolution/low-resolution feature maps extracted from earlier/later stage of a deep neural network are crucial to localize small/large text respectively. To effectively combine multi-level information to better capture widely varied appearances of video text, we propose an effective neural network model for video text detection based on a novel refine block structure, which exploits low-level high-resolution features from earlier convolutions to refine high-level low-resolution semantic features, and introduces residual connections to facilitate gradient propagation for efficient end-to-end training.

2.1. Network Architecture

The proposed text detection network is composed of three main parts: feature extraction branch, feature fusion branch and prediction module, as illustrated in Fig. 1.

2.1.1. Feature Extraction Branch

We adopt the ResNet-50 [10] network pretrained on ImageNet as the feature extraction branch for its strong representability. As shown in Fig. 1, the feature extraction branch is composed of interleaving convolution and pooling layers, which are organized into four blocks. Each block $i \in \{1, 2, 3, 4\}$ generates a set of feature maps f_i, whose sizes are $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$ of the input image, respectively.

2.1.2. Feature Fusion Branch

The feature fusion branch merges location information extracted from high-resolution layers with semantic information extracted from low-resolution layers. As shown in Fig. 1, the feature fusion branch is composed of a series of refine blocks proposed in this work. As one of its inputs, a refine block (denoted by Refine-i) connects to the output of the block-i of ResNet through one convolutional layer (with 1×1 filters), which reduces the number of channels in the feature maps f_i before inputting them into Refine-i. Meanwhile, the refine block Refine-i (except the bottom Refine-4) takes the high-level information outputted by Refine-i+1 as another input, and combines it with the high-resolution but low-level features from ResNet block-i. As the last step, a 3×3 convolution operation is performed on the final feature maps h_1, and the results are then fed to the prediction module. The refine block will be described in detail in Section 2.2.

2.1.3. Prediction Module

Taking the final feature maps of the feature fusion branch as input, the prediction module outputs 1-channel score map F_s and 5-channels geometry map F_g, among which 4 channels contain the parameters of the axis-aligned bounding box $\mathbf{R}$ of the text and 1 channel contains its rotation angle θ. We adopt the scheme in [6] to generate appropriate labels for the corresponding prediction channels.

To eliminate false and redundant text candidates in the output maps, we first apply thresholding on the output score of every candidate and discard those with lower scores. Then, Non-Maximum Suppression (NMS) is performed on the candidates to yield the final detection results.

2.2. Refine Block

We propose a novel refine block structure as the basic network unit of feature fusion branch, which uses the high-resolution features extracted by ResNet to refine the low-resolution semantic features from the preceding refine block.

As illustrated in Fig. 2, the structure of one refine block is composed of three components: residual unit, multi-resolution fusion and residual inception module. Note that, Refine-4 has only one input, while other refine blocks in the proposed detection network have two inputs.

2.2.1. Residual Unit

To adapt the pretrained ResNet network used in the feature extraction branch to our text detection task, we pass each input path of the refine block through one corresponding residual unit, in which we remove the batch-normalization layers used by ResNet [10] in its convolution units.

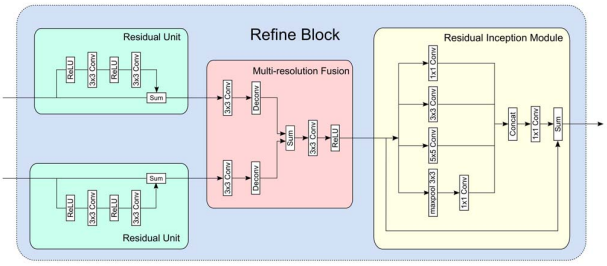

Fig. 2. The architecture of the refine block.

2.2.2. *Multi-resolution Fusion*

The multi-resolution fusion module fuses feature maps of all input paths to a refine block into high-resolution feature maps. It first applies 3×3 convolution on each input path for adaptation, and then deconvolves the resulting feature maps (usually with varied resolutions between different input paths) to feature maps with the highest input resolution. Finally, it adds all feature maps up and then performs 3×3 convolutions followed by ReLU operation in order to fuse the information from multi-resolution input paths. Note that, if there is only one input path to the refine block such as Refine-4 in Fig. 1, the input path will directly go through this part without changes.

2.2.3. *Residual Inception Module*

To enable multiple receptive fields of different sizes to effectively integrate features in different scales and allow capturing background context of a large image region, we propose to cascade a residual inception module after the fusion block as shown in Fig. 2. The residual connection in the residual inception module facilitates gradient propagation, which accelerates the end-to-end training of the model.

2.3. Loss Functions

The total loss L of the proposed model is the weighted combination of the classification loss L_s on the score map and the geometry loss L_g on the geometry map:

$$L = L_s + \lambda_g L_g \tag{1}$$

We set the weight factor $\lambda_g = 1.0$ in our experiment.

2.3.1. *Loss for Score Map*

We use the dice coefficient to calculate the classification loss L_s as follows:

$$L_s = 1 - \frac{2\sum_i^N \hat{\mathbf{Y}}_i \mathbf{Y}_i^*}{\sum_i^N \hat{\mathbf{Y}}_i + \sum_i^N \mathbf{Y}_i^*} \tag{2}$$

where, $\hat{\mathbf{Y}}$ and $\mathbf{Y}^*$ denote the predicted and the ground-truth values in the score map respectively, N denotes the number of elements in the score map.

2.3.2. *Loss for Geometry Map*

The geometry loss L_g is a weighted sum of the boundary loss L_b and the rotation angle loss L_θ as follows:

$$L_g = L_b + \lambda_\theta L_\theta \tag{3}$$

where $\lambda_\theta = 20$ is used in our experiment.

We adopt IoU loss for the boundary loss L_b due to its invariance against objects of different scales, and compute the rotation angle loss L_θ as follows:

$$L_b = -log\frac{|\hat{\mathbf{R}} \cap \mathbf{R}^*|}{|\hat{\mathbf{R}} \cup \mathbf{R}^*|} \qquad L_\theta(\hat{\theta}, \theta^*) = 1 - cos(\hat{\theta} - \theta^*) \tag{4}$$

where, $\hat{\mathbf{R}}$ and $\hat{\theta}$ denote the predicted boundary and rotation angle respectively, while $\mathbf{R}^*$ and θ^* denote corresponding ground truth.

2.4. Implementation Details

The number of filters used in every convolution layer and max-pooling layer in the proposed text detection network are labeled in Fig. 1, except for the refine blocks.

For one refine block illustrated in Fig. 2, the number of filters used in the convolution layers in the residual unit and the multi-resolution fusion module is equal to the number of filters used in the 1×1 convolution layer connecting the corresponding ResNet block to the refine block as shown in Fig. 1. In the residual inception module, we set the number of filters used in the last 1×1 convolution layer equal to the number of the input feature maps N_{ri}, and set the number of filters used in the other convolution layers to $N_{ri}/4$. The stride in all convolution and max-pooling layers is set to 1.

To speed up learning and augment the training data, we randomly sample 512×512 crops from training images to form a minibatch of size 14. The ADAM optimizer is exploited for training the proposed network. The learning rate starts from 10^{-4} and stops at 10^{-5} with a decay of 0.9 every 10000 steps.

3. TEXT TRACKING IN VIDEO

Due to the complexity of video context and the widely varied quality of video text, there usually exist some falsely detected candidates as well as missed text in the detection output. To enhance the final text detection performance, we propose to exploit the correlation of corresponding text cues in different video frames through a robust correlation filter based video text tracking algorithm as well as an effective combination mechanism for tracked and detected text candidates, which

has two main favorable effects: eliminating transient false detections and recovering missed text under temporarily unfavorable detection conditions based on the tracking result.

3.1. Tracking Based on Correlation Filter

We exploit the Staple [11] tracker to track text in the video. As a correlation filter based tracking algorithm, given an initial target object, the Staple tracker dynamically learns and updates a robust filter capturing the characteristics of the target object (depicted by HOG and color histograms), which is convolved over possible regions in the frame where the target may appear. The location with highest response is then regarded as the tracking result for the frame.

To initialize the Staple tracker, we compare each text candidate d detected in the current frame with all existing tracking trajectories (initially empty). If d overlaps with the tracked text candidate t ($IoU(t, d) > 0.8$ in this work) of an existing trajectory, we merge d with t, otherwise we create a new trajectory for text candidate d.

During the tracking, on the other hand, we look for the detected candidate d overlapping with the tracked candidate t of each trajectory in the current frame. If such d can be found, current frame is taken as a *matched* frame for the trajectory, otherwise it's denoted as a *unmatched* frame. If there have appeared 5 successive unmatched frames till the current frame in a trajectory, we terminate tracking t and its trajectory.

3.2. Combination of Tracking and Detection

Given all tracking trajectories of text in the video, which provide an important supplement to the detection results especially for frames with unfavourable detection conditions, we combine detected and tracked text candidates to improve the overall detection performance.

We first inspect the proportion of matched frames to all frames in each trajectory. If it falls below 0.7, the trajectory is regarded as invalid and all text candidates in it are discarded. By this mean, most transient falsely detected text candidates like noises will be eventually eliminated due to non-existence of matching tracked candidates in subsequent processing.

Next, we merge the text candidates $\mathbf{D}$ found by the text detector and those $\mathbf{T}$ localized by the tracker in each frame into the final set of text $\mathbf{X}$ using the following algorithm, which seeks an optimal matching between $\mathbf{D}$ and $\mathbf{T}$ so as to help recover text candidates missed by the detector:

1. Compute matching cost matrix $\mathbf{C} \in \mathbb{R}^{|\mathbf{T}|\times|\mathbf{D}|}$ between $\mathbf{T}$ and $\mathbf{D}$.
2. Run *Hungarian Algorithm* on $\mathbf{C}$ to obtain the set of optimally matched pairs α between $\mathbf{T}$ and $\mathbf{D}$.
3. For each pair $(t, d) \in \alpha$ ($t \in \mathbf{T}$, $d \in \mathbf{D}$), compute their overlapping ratio $ro(t, d)$ defined as the overlapping area divided by the area of the smaller one.
   - If $ro(t, d) > \mathcal{T}_{ro}$ ($\mathcal{T}_{ro} = 0.3$ in this work), d is adopted as the final text.
   - Otherwise, both t and d are preserved as final text.
4. For unmatched candidates u left behind by Hungarian algorithm:
   - If u is a tracked one, preserve u as a recovered text candidate by the tracker.
   - If u is a detected one, discard u as a transient false detection.

The matching cost matrix $\mathbf{C}_{t,d}$ (t, d denoting a pair of tracked and detected text candidate respectively) is computed by integrating the spatial and appearance dissimilarities (specifically on the position, color histogram and HOG features) between t and d:

$$\mathbf{C}_{t,d} = \mathbf{c}(\mathbf{t_r}, \mathbf{d_r}) \times \mathbf{c}(\mathbf{t_c}, \mathbf{d_c}) \times \mathbf{c}(\mathbf{t_g}, \mathbf{d_g}) \tag{5}$$

where, $\mathbf{c}(\mathbf{t}, \mathbf{d}) = \exp(\frac{\mathbf{dist}(\mathbf{t},\mathbf{d})}{\sigma})$, $\mathbf{dist}(\cdot, \cdot)$ is the Euclidean distance. $\mathbf{t_r}$/$\mathbf{d_r}$, $\mathbf{t_c}$/$\mathbf{d_c}$ and $\mathbf{t_g}$/$\mathbf{d_g}$ are the centroid coordinate, color histogram and HOG feature of t/d, respectively. σ denotes the normalization scale factors for the spatial, color histogram and HOG feature distances respectively.

4. EXPERIMENTS

4.1. Dataset

We adopt the dataset of ICDAR 2013 Robust Reading Competition Challenge 3: Text in Videos [12] and the dataset presented in [13] for video text detection performance evaluation.

The ICDAR 2013 dataset consists of 13 training videos and 15 testing videos, each lasting from 10 seconds to 1 minute. The dataset [13] consists of 5 videos of varied outdoor scenes, lasting from 6 seconds to 41 seconds. Specifically, because dataset [13] does not provide or specify the training set, we use the training images of ICDAR 2013 [12] and ICDAR 2015 [14] Robust Reading Competitions as the training set for experiments on dataset [13], which includes a total of 1229 images.

We adopt the standard evaluation metrics, precision p, recall r and f-measure, to evaluate the text detection performance of one method.

4.2. Evaluation of Text Detection

Table 1 shows the ablation experiment results of the proposed fundamental refine block structure of the detection model on individual video frames. It can be seen that, by integrating each proposed component into the network, the detection performance is consistently improved on both datasets.

To further investigate the effectiveness of the proposed text detection model, Table 1 compares its performance with

Table 1. Contribution of each main component of the proposed refine block structure to text detection performance (%). (a) Multi-resolution fusion, (b) Residual unit, (c) Residual inception module.

Method	Dataset [13]			ICDAR 2013 Dataset		
	p	r	f	p	r	f
(a)	83.5	85.2	84.0	67.2	55.3	58.5
(a)+(b)	85.4	86.2	85.4	68.2	55.5	58.9
(a)+(b)+(c) (**Proposed**)	**85.8**	**87.9**	**86.3**	**68.3**	**56.2**	**59.4**
EAST [6]	81.6	83.0	81.8	65.2	54.6	57.7

Table 2. Comparison of video text detection performances (%) with and without text tracking.

Method	Dataset [13]			ICDAR 2013 Dataset		
	p	r	f	p	r	f
w/o tracking	85.8	87.9	86.3	**68.3**	56.2	59.4
w. tracking	**90.0**	**90.2**	**89.2**	67.4	**56.9**	**59.8**

that of the popular EAST network [6] trained on the aforementioned datasets, which also exploits FCN framework. Compared to EAST, the proposed model achieves overall improved text detection performance - $4.2\%/3.1\%$ on precision, $4.9\%/1.6\%$ on recall, $4.5\%/1.7\%$ on f-measure, on dataset [13] and ICDAR 2013 dataset respectively.

4.3. Evaluation of Text Tracking

To evaluate the effect of the text tracking mechanism on improving text detection results, Table 2 compares the final detection performances with and without the tracking mechanism. It can be seen that, combining tracking with text detection significantly enhances all performance scores on dataset [13], by filtering out transient false text candidates detected and recovering temporarily missed text. On ICDAR 2013 dataset, on the other hand, since it contains more complicated circumstances interfering the tracking such as densely clustered text and drastic camera movements, which result in unstable tracking results and thereby the relatively small improvements on recall and f-measure and the slight reduction of the precision after combining tracking with detection.

In Fig. 3, we illustrate how text tracking helps improve the results of individual-frame detection. In example (a), the temporary highlight on the text hinders the detector from acquiring the complete bounding box of the whole text, which is recovered based on tracking results. In example (b), the detector returns a false text candidate (actually some windows of a building) near the top of the image because of its text-alike appearance, which is discarded by tracking due to its temporally over-short tracking trajectory.

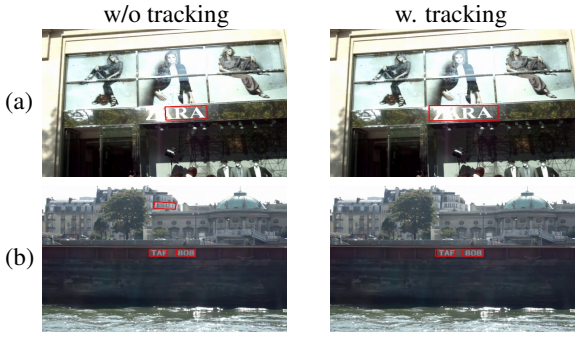

Fig. 3. Illustration of the effect of text tracking on improving text detection in the video.

Table 3. Comparison of video text detection performances (%) of the proposed method and some state-of-the-art methods.

Dataset [13]

Method	p	r	f
Proposed	**90.0**	**90.2**	**89.2**
Wang *et al.* [9]	88.8	87.5	88.1
Yang *et al.* [7]	85	77	81
Zuo *et al.* [15]	84	68	75
Minetto *et al.* [13]	61	60	63

ICDAR 2013 Dataset

Method	p	r	f
Proposed	**67.4**	**56.9**	**59.8**
Wang *et al.* [9]	58.3	51.7	54.5
Khare *et al.* [16]	57.9	55.9	51.7
Yin *et al.* [2]	48.6	54.7	51.6
Shivakumara *et al.* [17]	51.2	53.7	50.7
Zhao *et al.* [8]	47.0	46.3	46.7
Epshtein *et al.* [1]	39.8	32.5	35.9

4.4. Comparison with State-of-the-Art Methods

We compare the performances of the proposed video text detection method and some state-of-the-art methods in Table 3.

The proposed method achieves the highest precision, recall and f-measure among all methods being compared on both datasets, which demonstrate the effectiveness of the proposed text detection network and tracking mechanism. Specifically, on the more complicated ICDAR 2013 dataset where scene text appearance and context vary significantly, the proposed method achieves enhanced scores on all performance metrics compared to the second best scores - 9.1% increase on precision, 1.0% increase on recall and 5.3% increase on f-measure. Fig. 4 shows some video text detection results by the proposed method, which exhibits well robustness to the widely varied qualities of text in the videos.

Fig. 4. Examples of video text detection results by the proposed method.

5. CONCLUSIONS

We present an effective fully convolutional network model for detecting text in videos based on a novel refine block structure, which refines the low-resolution semantic features from one input path with the high-resolution low-level features from another input path. The model hierarchically combines multiple refine blocks and corresponding feature extraction blocks to capture widely varied appearances of video text with refined features that integrate multi-resolution information. We further complement the individual-frame detection with an efficient correlation filter based text tracking mechanism, and effectively enhance the overall detection performance by combining tracking and detection results.

6. REFERENCES

[1] Boris Epshtein, Eyal Ofek, and Yonatan Wexler, "Detecting text in natural scenes with stroke width transform," in *CVPR*, 2010, pp. 2963–2970.

[2] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao, "Robust text detection in natural scene images," *IEEE TPAMI*, vol. 36, no. 5, pp. 970–983, May 2014.

[3] Shangxuan Tian, Yifeng Pan, Chang Huang, Shijian Lu, Kai Yu, and Chew Lim Tan, "Text flow: A unified text detection system in natural scene images," in *ICCV*, 2015, pp. 4651–4659.

[4] Baoguang Shi, Xiang Bai, and Serge Belongie, "Detecting oriented text in natural images by linking segments," in *CVPR*, 2017, pp. 3482–3490.

[5] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao, "Detecting text in natural image with connectionist text proposal network," in *ECCV*, 2016, pp. 56–72.

[6] 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*, 2017, pp. 2642–2651.

[7] Chun Yang, Xu-Cheng Yin, Wei-Yi Pei, Shu Tian, Ze-Yu Zuo, Chao Zhu, and Junchi Yan, "Tracking based multi-orientation scene text detection: A unified framework with dynamic programming," *IEEE TIP*, vol. 26, no. 7, pp. 3235–3248, July 2017.

[8] 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 TIP*, vol. 20, no. 3, pp. 790–799, March 2011.

[9] Lan Wang, Yang Wang, Susu Shan, and Feng Su, "Scene text detection and tracking in video with background cues," in *ICMR*, 2018, pp. 160–168.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[11] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H. S. Torr, "Staple: Complementary learners for real-time tracking," in *CVPR*, 2016, pp. 1401–1409.

[12] 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*, 2013, pp. 1484–1493.

[13] Rodrigo Minetto, Nicolas Thome, Matthieu Cord, Neucimar J. Leite, and Jorge Stolfi, "Snoopertrack: Text detection and tracking for outdoor videos," in *ICIP*, 2011, pp. 505–508.

[14] "ICDAR 2015 robust reading competition," http://rrc.cvc.uab.es, 2015.

[15] Ze-Yu Zuo, Shu Tian, Wei yi Pei, and Xu-Cheng Yin, "Multi-strategy tracking based text detection in scene videos," in *ICDAR*, 2015, pp. 66–70.

[16] Vijeta Khare, Palaiahnakote Shivakumara, Raveendran Paramesran, and Michael Blumenstein, "Arbitrarily-oriented multi-lingual text detection in video," *Multimedia Tools and Applications*, vol. 76, no. 15, pp. 16625–16655, 2017.

[17] Palaiahnakote Shivakumara, Rushi Padhuman Sreedhar, Trung Quy Phan, Shijian Lu, and Chew Lim Tan, "Multioriented video scene text detection through bayesian classification and boundary growing," *IEEE TCSVT*, vol. 22, no. 8, pp. 1227–1235, Aug 2012.