Semantic-Aware Video Text Detection

Wei Feng[1,2] Fei Yin[1,2] Xu-Yao Zhang[1,2] Cheng-Lin Liu[1,2,3]
[1] National Laboratory of Pattern Recognition (NLPR),
Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[3] CAS Center for Excellence of Brain Science and Intelligence Technology, Beijing 100190, China
Email: {wei.feng, fyin, xyz, liucl}@nlpr.ia.ac.cn

Abstract

Most existing video text detection methods track texts with appearance features, which are easily influenced by the change of perspective and illumination. Compared with appearance features, semantic features are more robust cues for matching text instances. In this paper, we propose an end-to-end trainable video text detector that tracks texts based on semantic features. First, we introduce a new character center segmentation branch to extract semantic features, which encode the category and position of characters. Then we propose a novel appearance-semantic-geometry descriptor to track text instances, in which semantic features can improve the robustness against appearance changes. To overcome the lack of character-level annotations, we propose a novel weakly-supervised character center detection module, which ***only uses word-level annotated real images*** *to generate character-level labels. The proposed method achieves state-of-the-art performance on three video text benchmarks ICDAR 2013 Video, Minetto and RT-1K, and two Chinese scene text benchmarks CASIA10K and MSRA-TD500.*

1. Introduction

Video text detection aims to localize and track text instances in videos. It has attracted much attention in recent years, due to its wide application in video analysis and multimedia information retrieval. Although previous methods [34, 41, 52] have made significant efforts in both text detection and tracking, it is still a challenging task because of motion blur and illumination changes.

Most existing methods [56, 46, 34] treat text detection and tracking separately, where a single frame is detected first, then text tracking methods are applied based on detection results. However, these methods ignore the temporal contexts and the information interaction between detection and tracking. Recently, Yu *et al.* [52] proposed an end-to-end trainable framework to integrate text detection and tracking, in which the appearance-geometry descriptor is used to track text instances. However, the proposed descriptor is mainly based on the text appearance, which is easily influenced by the change of perspective and illumination. In contrast to appearance features, semantic features are robust cues for matching text instances. For example, most text instances fail to match due to large perspective changes as shown in Fig 1 (a). However, the character position and category of the same text instance from different perspectives are similar. When there are priori semantic features, wrong matching results could be corrected as shown in Fig 1 (b). Although both word-level and character-level annotations provide semantic information, character-level annotations contain more detailed structure information, which are

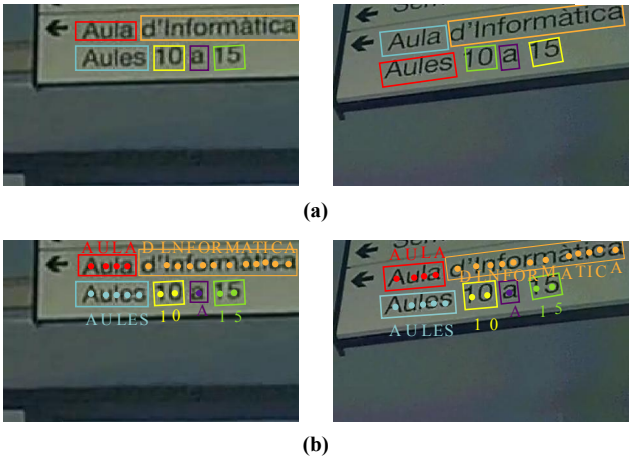

Figure 1. (a) The text appearance changes dramatically from different perspectives, which makes the tracking branch failed to match text instances. (b) The category and position of characters can help the tracking branch to match text instances more accurately. Boxes with the same color belong to the same trajectory, and dots represent the character centers.

more powerful references for text tracking. Unfortunately, character-level annotations of real datasets are too costly.

To generate character-level annotations of real datasets automatically, some methods [1, 44] were proposed in a weakly-supervised learning way. In these methods, a character detector is first trained on synthetic datasets, then the trained detector detects characters of real images. There are two main disadvantages of these methods. On one hand, there is a large domain gap between synthetic and real images, which makes the performance of character detector on real images unsatisfactory. On the other hand, the widely used synthetic datasets only focus on English, so it is difficult to transfer these methods to other languages without synthetic datasets.

To exploit the semantic information in video text detection while overcome the lacking of character-level annotated data, we propose a semantic-aware video text detection framework as shown in Fig 2, in which character-level annotations are directly generated from word-level annotated real datasets. Specifically, a ConvLSTM [30] block is used to propagate frame-level information, which makes full use of the temporal contexts in videos. Then, a character center segmentation task in the mask head of Mask R-CNN [9] is designed to encode the position and category of characters as semantic features. Based on appearance features and newly added semantic features, appearance-semantic-geometry descriptors ($ASGD$) are introduced to robustly represent text instances, which are matched with stored $ASGD$ of previous frames to achieve text tracking. Although the proposed framework needs character-level annotations, we adopt a sliding-window based text recognizer [38, 49] to detect character centers automatically, which only needs word-level annotated real images to train. This makes our framework easy to apply to multiple languages, such as Chinese which is typical of large character set. To the best of our knowledge, this is the first video text detector to introduce semantic features into text detection and tracking, which only uses word-level annotated real images to generate character-level labels.

Our contributions are in three folds: (1) We propose a novel end-to-end video text detector, which unifies text and character detection, and text tracking. (2) An appearance-semantic-geometry descriptor is proposed, in which the semantic features help improve the robustness to appearance changes. (3) Character-level annotations are generated in a weakly-supervised way, which improves the practicability of our method. The proposed method is effective for both text detection and tracking, and has achieved state-of-the-art performance on three video text datasets ICDAR 2013 Video [16], Minetto [25] and RT-1K [27], and two Chinese scene text datasets CASIA10K [11] and MSRA-TD500 [47].

2. Related Work

Text detection in videos usually combines a single frame text detector and some specific tracking techniques. Therefore, we review related works including single frame text detection and video text detection. For more details, please refer to the surveys of [51, 48, 55].

2.1. Single Frame Text Detection

Traditional methods [33, 10, 2] detect components of text first, then aggregate components into final detection results. The disadvantages of these methods lie in the error accumulation and inefficiency. Regression based methods [18, 12, 54] adopt similar ideas to generic object detection with some text-specific modifications. To detect arbitrary shaped text, some methods [39, 40, 23, 5] first detect local units, and then aggregate them into final results.

Recently, some methods use character-level annotations to provide detailed semantic information for text detection. Baek *et al.* [1] detected text instances by exploring each character and affinity between characters. Xing *et al.* [44] detected bounding boxes of words and characters directly in one pass. Liao *et al.* [17] added a character segmentation branch on the basis of Mask R-CNN. However, these methods need synthetic datasets to pre-train the character detector. Different from these methods, the proposed method generates character-level labels directly from real datasets, which has much more practical values.

2.2. Video Text Detection

Most video text detection methods are based on tracking with single frame detection results. Zuo *et al.* [56] proposed a multi-strategy text tracking method, which fuses the advantages of several tracking techniques. Tian *et al.* [34] proposed a unified tracking based text detection system with dynamic programming. Yang *et al.* [46] tracked proposals in adjacent frames with a motion-based method. However, these methods ignore the temporal contexts in the video.

To capture spatial-temporal information, Wang *et al.* [37] made use of the temporal correlation of text cues across successive frames. Yu *et al.* [52] used ConvLSTM to catch long-term spatial-temporal information. Although these methods have made great progress, the tracking branch is mainly based on text appearance features, which are sensitive to the changes in appearance. The proposed method adopts an appearance-semantic-geometry descriptor, making the framework robust to appearance changes.

3. Methodology

An overview of the proposed end-to-end video text detector is illustrated in Fig 2. After extracting visual features by the stem network, a ConvLSTM block is used to extract

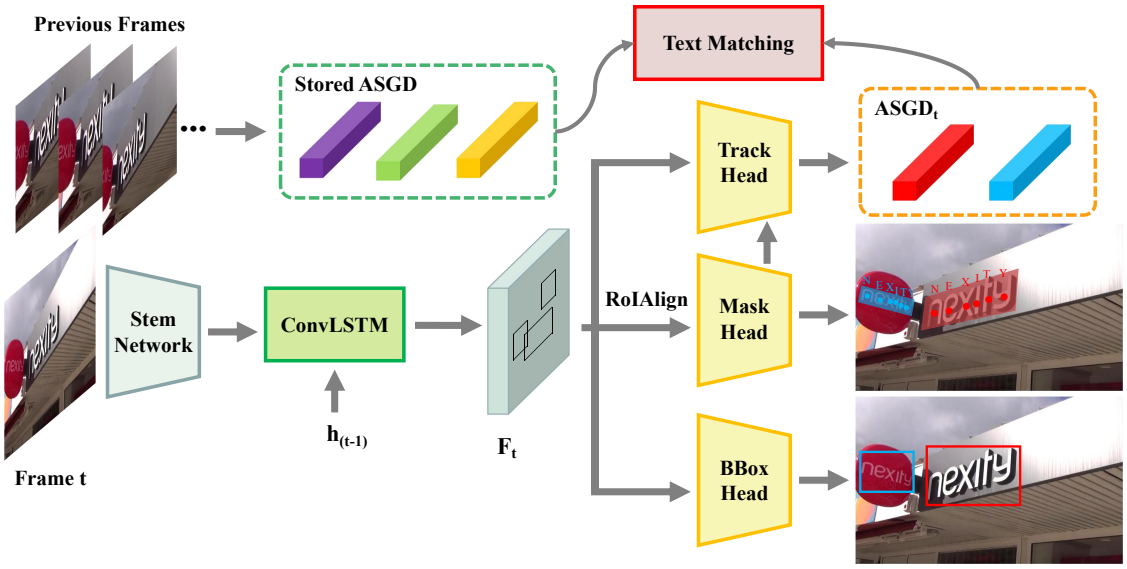

Figure 2. An overview of the proposed framework. A character center segmentation task is embedded in the mask head of Mask R-CNN to extract semantic features, making appearance-semantic-geometry descriptors (ASGD) robust to appearance changes.

spatial-temporal information. Then, we embed a character center segmentation task in the mask head to localize and recognize characters, which can extract semantic features. Finally, the text tracking head generates appearance-semantic-geometry descriptors, which are matched with detected text instances of previous frames. In addition, a sliding-window based text recognizer is introduced to provide character-level labels for the character center segmentation task. The text recognizer can localize character centers in a weakly-supervised way. In the following, we will describe the details of text detection, text tracking, weakly-supervised character detection and inference procedure.

3.1. Text Detection

Different from scene images, videos always contain abundant temporal information. Therefore, we adopt a ConvLSTM block to integrate long-term temporal information. Denote the visual features extracted by the stem network at t-th frame as V_t. The output F_t of the ConvLSTM block can be formulated as:

$$(F_t, h_t) = ConvLSTM(V_t, h_{(t-1)}), \quad (1)$$

where h_t and $h_{(t-1)}$ represent the hidden states at time t and $t-1$, respectively. In this way, features can propagate frame-level information in a long time range.

After integrating temporal information, we adopt Mask R-CNN to predict axis-aligned rectangular bounding boxes and the corresponding instance segmentation masks, which consists of two stages. First, a region proposal network (RPN) [28] is used to propose a set of candidate text regions of interest (RoIs). Second, the RoIAlign [9] operation extracts features from F_t within each RoI, then the extracted features are used to perform classification, bounding box regression and instance segmentation. As Mask R-CNN can detect arbitrary shaped texts in an instance segmentation manner, we fit a minimum enclosing rotated rectangle to each mask for oriented texts.

To enhance the detection performance and extract semantic features for the following tracking head, we add a character center segmentation branch on the basis of Mask R-CNN. This branch has two convolution layers with 3×3 filters and a upsample layer with stride 2. Then the feature maps are used to generate final segmentation maps with channel S, where S is the number of character classes plus background category. For each character center, we regard pixels around the center within a distance r as positive. The parameter r is proportional to the shortest side of text boundaries by a ratio of 0.2. Then, we generate the ground truth map C^* by drawing the expanded character center regions on a zero-initialized mask and filling the regions with their corresponding category indexes. Denote the number of pixels in C^* as N. The loss function of character center segmentation is a weighted spatial softmax loss, which can be formulated as:

$$L_{char} = -\frac{1}{N}\sum_{n\in N} W_n \sum_{s\in S} C^*_{n,s} log(\frac{e^{C_{n,s}}}{\sum_{k\in S} e^{C_{n,k}}}), \quad (2)$$

where C represents output maps and W is a weighted matrix to balance the positive and negative loss. Denote the number of positive and negative pixels as N_{pos} and N_{neg}. The weight of positive pixels is 1, and the weight of negative pixels is N_{pos}/N_{neg}.

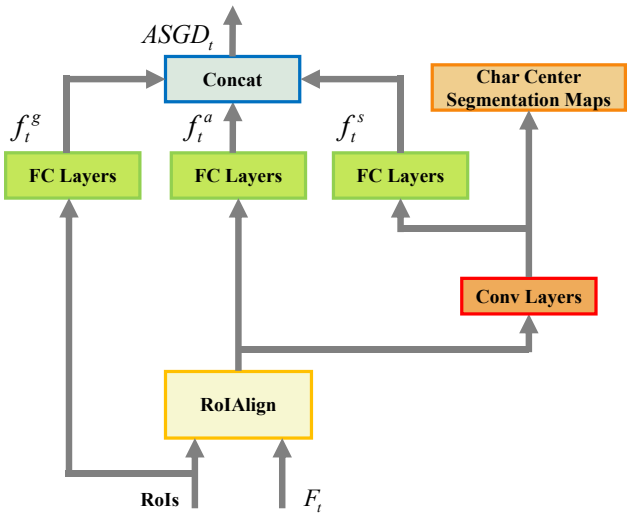

Figure 3. The proposed descriptor $ASGD_t$ consists of appearance features f_t^a, semantic features f_t^s, and geometry features f_t^g.

Combining the character center segmentation loss, the loss function of text detection can be calculated as:

$$L_{det} = L_{rpn} + \alpha_1 L_{rcnn} + \alpha_2 L_{mask} + \alpha_3 L_{char}, \quad (3)$$

where L_{rpn}, L_{rcnn} and L_{mask} represent loss functions of RPN, Fast R-CNN [28] and instance segmentation, respectively. α_1, α_2 and α_3 are set to 1 in our experiments.

Mask TextSpotter v1 [24] and v2 [17] also combine character segmentation task with the original Mask R-CNN. However, these methods need character-level annotated synthetic and real images for training. Different from Mask TextSpotter, the character-level annotations used in our detector are all generated from only word-level annotated real data, which will be described in Section 3.3.

3.2. Text Tracking

Previous methods perform text tracking with appearance features extracted from text RoIs. However, the rough appearance features make text tracking easily influenced by the change of perspective and illumination. Instead of only considering text appearance features, we argue that semantic features could provide robust prior information for text tracking. Therefore, we encode the position and category of characters as part of input for the text tracking task. To represent text instances robustly, we propose a novel appearance-semantic-geometry descriptor ($ASGD$), which consists of three parts as shown in Fig 3. First, we employ RoIAlign layer to extract features from F_t within text RoIs, then two fully connection layers are used to project the extracted features into new ones. We call the new features as text appearance features f_t^a. Second, we also use two fully connection layers to project the intermediate features from the second convolution layer of character segmentation branch into semantic features f_t^s, which encode the position and category of characters. Third, the coordinates of text RoIs are embedded as geometry features f_t^g. Finally, these three parts are concatenated to generate the descriptor $ASGD_t$, which can be formulated as follows:

$$ASGD_t = Concat([f_t^a, f_t^s, f_t^g]). \quad (4)$$

To train the text tracking branch, we use a pair of frames in which one frame is picked as the query frame and the other is the reference frame. For the query frame, we extract features within text RoIs which have at least 70% IoU with ground truth boxes. For the reference frame, we directly use ground truth boxes to extract features without generating text RoIs. To match text instances belonging to the same object, we follow the similar idea in [52], which makes descriptors close for positive pairs and far for negative pairs. However, the distances between positive pairs are difficult to approach 0 because of the difference caused by motion. Therefore, we adopt a smoothed double-margin loss based on the contrastive loss [8]. Denote the distances between $ASGD$ of query and reference frames as d. The loss function of text tracking can be formulated as:

$$L_{track} = y(R(d - m_p))^2 + (1 - y)(R(m_n - d))^2, \quad (5)$$

where R denotes the ReLU function. m_p and m_n are the margins for positive pairs and negative pairs, respectively. We set m_p as 0.3 and m_n as 1.0 in our implementation. Moreover, y is the pairs label whose value is 1 for positive pairs and 0 for negative pairs.

For end-to-end training of text detection and tracking, the whole loss function can be written as:

$$L = L_{det} + \beta L_{track}, \quad (6)$$

where β is the hyper-parameter to control the balance between detection and tracking. We set it as 0.5 in our experiments.

3.3. Weakly-Supervised Character Detection

As character-level annotations require much more human labeling efforts, previous methods usually generate character-level labels using synthetic datasets. However, synthetic datasets are mainly focused on English, and there is a large domain gap between synthetic and real images. Therefore, we propose a weakly-supervised character detection module to provide character-level labels for the character center segmentation task, which only needs word-level annotated real images. The pipeline of generating character-level labels on the training set is shown in Fig 4. First, we use the RoIRotate [22] operator to transform text instances into axis-aligned ones. Then we adopt a sliding-window based text recognizer [38, 49] to classify each window. When characters are located in the center of sliding

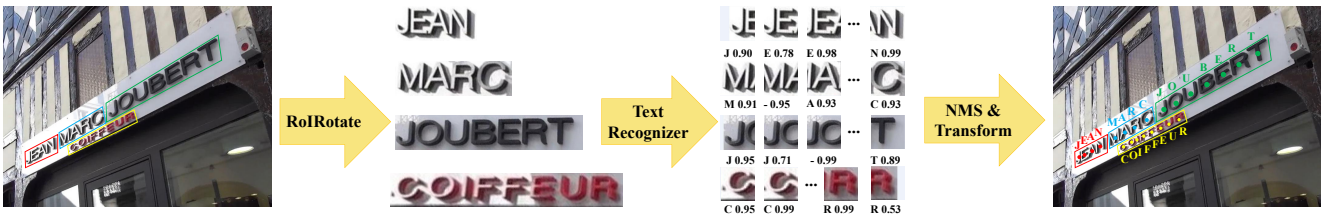

Figure 4. The pipeline of generating character-level labels. In the recognition results, the first item is the classification result, and the second item is the score. "-" means blank. We only show part of sliding windows for better visualization.

Table 1. The architecture of the sliding-window based text recognizer. Each convolution layer is followed by a batch normalization layer and a ReLU layer. S is the number of character classes, which is 37 for English datasets and 7357 for Chinese datasets.

Type	Configurations
Input	N×32×32
3×Conv_bn_relu	3×3, 100, stride 1×1
Max Pooling	2×2, stride 2×2
3×Conv_bn_relu	3×3, 150, stride 1×1
Max Pooling	2×2, stride 2×2
3×Conv_bn_relu	3×3, 300, stride 1×1
Max Pooling	2×2, stride 2×2
3×Conv_bn_relu	3×3, 400, stride 1×1
Max Pooling	2×2, stride 2×2
Fully Connection	256, drop: 0.5
Fully Connection	256, drop: 0.0
Fully Connection	S

windows, the text recognizer can recognize the character with high scores. When the centers of sliding windows and characters are misaligned, the text recognizer will output the blank labels or low scores. Finally, we conduct Non-Maximum Suppression (NMS) on the sliding windows, and transform the picked sliding window centers back to the input image as character center labels.

To train the text recognizer, we first transform text instances of the training set into axis-aligned ones, whose height is normalized as 32. Then we slide windows on the transformed text instance with stride l. We set l as 2 for English datasets and 4 for Chinese datasets. Finally, a VGG-like [32] network takes the sliding windows as input, and classifies each window. The architecture of text recognizer is shown in Table 1. To decode the label distribution to the final sequence, we adopt the Connectionist Temporal Classification [7] (CTC) decoder and assume that each sliding window represents a time step. Denote the CTC path as π and the mapping function as B. The conditional probability of the ground truth y^* is the sum of the probabilities of all the paths by B:

$$P(y^*|X) = \sum_{\pi \in B^{-1}(y^*)} P(\pi|X). \tag{7}$$

The objective is to maximize the log likelihood of Eq.7. The loss function of text recognition is formulated as:

$$L_{rec} = -\log p(y^*|X). \tag{8}$$

Although the text recognizer is easy to fit small datasets, it is hard to achieve satisfactory performance on large datasets, especially when the distribution of character classes is unbalanced. Therefore, we propose an iterative training process to improve the performance on the training set. We utilize a simple rule that identifies character center detection results as "correct" if the text recognition result is the same as the ground truth. The proposed iterative training process is described as follows.

(i) We first train an initial text recognizer on the whole training set until the loss becomes stable. Then we test the trained model on the same training set.

(ii) According to the previous rule, we collect text instances with correct recognition results to build character-level labels, and remove them from the training set. The trained text recognizer continues to be trained on the reduced training set.

(iii) This training process is performed iteratively to improve the character center detection results. As the number of iterations increase, the text recognizer can pay more attention to hard samples and rare characters as shown in the experiments.

3.4. Inference

The proposed method generates text detection results and matches the detected text instances in an online fashion. Given a frame of time step t, we first detect all text instances and obtain the corresponding $ASGD_t$ as Eq.4. Then, we calculate the similarity matrix between $ASGD_t$ and the stored $ASGD$ of previously detected text instances. Finally, we use Kuhn-Munkres algorithm with threshold value θ_m to get the matching pairs. If the text instance finds a matching text instance, we update this tracklet set and corresponding $ASGD$ stored in the memory. Note that we only save the latest $ASGD$ for one tracklet set. For no-matching text instances, we build new trajectories for them, and insert their $ASGD$ to the memory. Altogether, the proposed method can reach 9.6 fps on ICDAR 2013 Video dataset.

Figure 5. Examples of text detection and tracking results. First and second rows: video text detection. Boxes with the same color belong to the same trajectory. Third row: single frame text detection.

4. Experiments

We evaluate the text detection and tracking performance on three English video datasets. Since there is no public non-English video dataset, we evaluate our method on two Chinese scene image datasets to verify the applicability of our method on non-English datasets.

4.1. Datasets

ICDAR 2013 Video. This dataset contains 13 videos for training and 15 videos for testing, which are harvested from indoors and outdoors scenarios. The resolution ranges from 720×480 to 1280×960. Besides, each text is labeled as a quadrangle with 4 vertexes in word-level.

Minetto. The Minetto dataset has 5 videos in outdoor scenes. The resolution is fixed as 640×480. Each text is labeled in the form of axis-aligned bounding box. As all videos are for testing, we use the model trained on ICDAR 2013 Video to evaluate on this dataset directly.

RT-1K. The RT-1K dataset contains 1000 videos in road scenes, including 700 for training and 300 for testing. We evaluate our method on this dataset to verify the superiority of the proposed method on large scale video text datasets.

CASIA10K. This dataset is a large scale Chinese scene text dataset, which contains 7000 training images and 3000 testing images. As there is no widely used Chinese synthetic datasets, previous methods are difficult to obtain character-level labels.

MSRA-TD500. The MSRA-TD500 dataset consists of 300 training images and 200 training images. This dataset is focused on Chinese and English texts, where each text instance is labeled in line-level.

4.2. Implementation Details

The proposed method is implemented in PyTorch, and runs on a regular workstation with Nvidia Titan Xp. We adopt ResNet-50-FPN [9] as the stem network, which is pre-trained on ImageNet dataset [15]. The configuration of Mask R-CNN follows the public implementation on MS COCO [21]. We train the model in 12 epochs. The initial learning rate is set to 0.03, then the learning rate is decayed to a tenth at epoch 8 and 11. At test time, the shorter sides of input images are resized to 800 pixels.

The input images of sliding-window based text recognizer are scaled to height of 32 pixels with the aspect ratio unchanged. Then the width is padded to 512 for parallel training. We train the text recognizer with the initial learning rate as 0.1, and decrease the learning rate by ×0.3 at epoch 50 and 80. In the iterative training stage, we fix the learning rate as 0.009, and finish the training when the loss becomes stable. The number of training stages is three.

Table 2. Video text detection results on ICDAR 2013 test set. "W/o sf" represents without semantic features.

Method	Precision	Recall	F-measure
Epshtein *et al.* [4]	39.80	32.53	35.94
Zhao *et al.* [53]	47.02	46.30	46.65
Yin *et al.* [50]	48.62	54.73	51.56
Khare *et al.* [14]	57.91	55.90	51.70
Wang *et al.* [37]	58.34	51.74	54.45
Shivakumara *et al.* [31]	61.00	57.00	59.00
Wang *et al.* [42]	71.90	58.67	62.65
Wu *et al.* [43]	63.00	**68.00**	65.00
Yu *et al.* [52]	**82.36**	56.36	66.92
Our two-stage	66.81	63.92	65.33
W/o sf	67.53	65.58	66.54
Proposed	75.46	64.08	**69.31**

Table 3. Video text detection results on Minetto test set. "W/o sf" represents without semantic features.

Method	Precision	Recall	F-measure
Minetto *et al.* [25]	61.00	69.00	63.00
Zuo *et al.* [56]	84.00	68.00	75.00
Tian *et al.* [33]	85.00	77.00	81.00
Wang *et al.* [42]	83.03	84.22	83.30
Yang *et al.* [46]	89.00	84.00	86.00
Wang *et al.* [37]	88.80	87.53	88.14
Yu *et al.* [52]	91.27	89.38	90.32
Our two-stage	91.77	88.68	90.19
W/o sf	93.21	89.53	91.34
Proposed	**96.90**	**91.32**	**94.02**

Table 4. Video text detection results on RT-1K test set. "W/o sf" represents without semantic features. Results other than ours are obtained from [27].

Method	Precision	Recall	F-measure
CTPN [35]	0.44	0.41	0.42
EAST [54]	0.42	0.30	0.35
FOTS [22]	0.45	0.36	0.40
W/o sf	0.73	0.42	0.53
Proposed	**0.76**	**0.43**	**0.55**

Table 5. Video text tracking results on Minetto test set. "MOTP" and "MOTA" represent Multi-Object Tracking Precision and Multi-Object Tracking Accuracy, respectively. "W/o sf" represents without semantic features.

Method	MOTP	MOTA
Zuo *et al.* [56]	73.07	56.37
Pei *et al.* [26]	73.07	57.71
Geometry descriptor [52]	76.66	74.04
Matching AGD with AGD [52]	74.70	75.62
Matching AGD with EAGD [52]	75.72	81.31
Our two-stage	75.32	78.71
W/o sf	75.33	80.26
Proposed	**76.78**	**83.53**

Table 6. Detection results on CASIA10K test set. "W/o sf" represents without semantic features. Results other than ours are obtained from [11].

Method	Precision	Recall	F-measure
EAST [54]	77.71	53.27	63.21
SegLink [29]	72.75	69.67	71.18
He *et al.* [11]	81.28	70.48	75.50
Feng *et al.* [6]	86.34	72.60	78.88
W/o sf	75.42	81.22	78.21
Proposed	**76.55**	**84.10**	**80.15**

4.3. Comparison with the State-of-the-art

We compare the performance with previous works on several datasets to verify the superiority of our method.

Table 7. Detection results on MSRA-TD500 test set. "W/o sf" represents without semantic features.

Method	Precision	Recall	F-measure
PixelLink [3]	83.0	73.2	77.8
RRD [20]	87.0	73.0	79.0
Xue *et al.* [45]	83.0	77.4	80.1
CRAFT [1]	88.2	78.2	82.9
Tian *et al.* [36]	84.2	**81.7**	82.9
DB [19]	**91.5**	79.2	84.9
W/o sf	87.8	79.4	83.3
Proposed	89.2	81.5	**85.2**

Table 8. Detection results on ICDAR 2015 test set. "P", "R", "F" represent "Precision", "Recall", "F-measure", respectively.

Method	P	R	F
Mask TextSpotter v1 [24]	91.6	81.0	86.0
CRAFT [1]	89.8	84.3	86.9
Mask TextSpotter v2 [17]	86.6	87.3	87.0
CharNet [44] R-50	88.3	91.1	89.7
Proposed	88.5	85.8	87.1

4.3.1 Video Text Detection

Our method achieves state-of-the-art performance on three video text datasets as shown in Tables 2, 3, 4 and 5. With the help of semantic features, our method is robust to the change of perspective and illumination, and outperforms previous methods in both text detection and tracking tasks. It should be noticed that character-level annotations used in the training stage are generated in a weakly-supervised manner, which owns more practical values. Some qualitative results are shown in Fig 5.

4.3.2 Single Frame Text Detection

Our method also reaches state-of-the-art performance on two Chinese scene text datasets as shown in Tables 6 and 7. The single frame detector is built by removing the ConvLSTM block and the text tracking loss. As the proposed method only needs word-level annotated real images, it is easy to apply on non-English datasets. We also compare the detection performance on an English scene text dataset ICDAR 2015 [13] with other character based methods as shown in Table 8. Our method achieves competitive results to the state-of-the-art approaches, which need synthetic datasets to generate character-level labels. This shows the superiority of our method. Some single frame text detection results are shown in Fig 5.

4.4. Ablation Studies

We conduct some comparison experiments to verify the benefits of semantic features, iterative training process and end-to-end training.

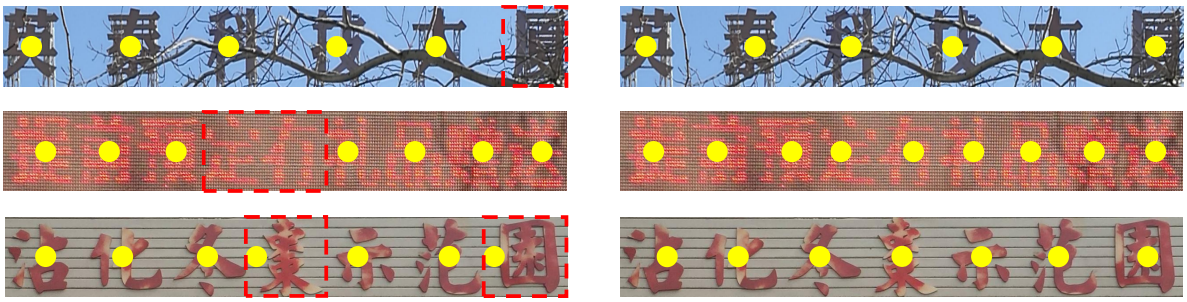

Figure 6. Iterative training process can improve character center detection performance on large scale datasets. From left to right: initial and final detection results. Yellow dots show character center detection results. Red dashed boxes show that the initial text recognizer is difficult to detect noisy, obscure and rare characters.

Table 9. Iterative training improves character detection performance. "Line accuracy" is evaluated on the whole training set of CASIA10K, and "Detection accuracy" is evaluated on 500 images labeled by ourself. At the step 0, the text recognizer is trained in 100 epochs, and 20 epochs for other steps.

Step	Line accuracy	Detection accuracy	Epochs
0	77.2	70.1	100
1	82.3	83.1	20
2	96.4	95.2	20

4.4.1 Effect of Semantic Features

The position and category of characters can provide robust semantic features for text tracking and detection. Without semantic features, text tracking may be influenced by the appearance changes. Meanwhile, the text detector may ignore text instances that are not salient. To demonstrate the benefits of semantic features, we evaluate a variant of our method which removes the character center segmentation loss, and the descriptors in text tracking branch only consist of appearance and geometry features. As shown in Tables 2, 3 4 and 5, the proposed method outperforms the one without semantic features in both text detection and tracking. We also report the performance without semantic features on Chinese datasets as shown in Tables 6 and 7, which verifies that semantic features are beneficial for both Chinese and English.

4.4.2 Effect of Iterative Training

The proposed iterative training process aims to enhance the character center detection performance gradually, especially when the character distribution is unbalanced. To demonstrate the importance of iterative training, we label the character centers of 500 images from CASIA10K dataset, and evaluate the character detection performance of each iteration. As shown in Table 9, the initial line accuracy and character detection performance are poor due to large and unbalanced character categories. As the number of iterations increases, the line accuracy and character detection performance increase continuously. After three iterative steps, the line accuracy and character detection performance exceed 95%, which allows us to train the character center segmentation branch with only word-level annotated real images. We also show some qualitative examples in Fig 6.

4.4.3 Effect of End-to-End Training

Most previous methods perform text detection and tracking separately, which ignore the correlation between two tasks. Unlike these methods, the proposed method unifies text detection and tracking in an end-to-end framework. To verify the effect of end-to-end training, we evaluate a variant of our method which trains text detection and tracking separately. As shown in Tables 2, 3 and 5, the proposed method outperforms our two-stage method by a large gap, which demonstrates that these two tasks can benefit each other.

5. Conclusion

In this paper, we propose a novel semantic-aware video text detector, by incorporating semantic information to improve detection and tracking performance. The text detector detects text instances and character centers simultaneously, which can extract semantic features. With the help of semantic features, the text tracking branch is more robust to the appearance changes. Furthermore, we propose a sliding-window based text recognizer to generate character-level labels from word-level annotated real datasets, which avoids the requirement and disadvantages of synthetic data. Experiments on several datasets have demonstrated the effectiveness of our method. A future improvement would be to combine multi-level semantic features for coping with more complicated scene videos.

Acknowledgement

This work has been supported by the National Natural Science Foundation of China (NSFC) grants 61733007 and 61721004.

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9365–9374, 2019. 2, 7

[2] Huizhong Chen, Sam S Tsai, Georg Schroth, David M Chen, Radek Grzeszczuk, and Bernd Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 2609–2612, 2011. 2

[3] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In *AAAI Conference on Artificial Intelligence*, 2018. 7

[4] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2963–2970, 2010. 6

[5] Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Textdragon: An end-to-end framework for arbitrary shaped text spotting. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9075–9084, 2019. 2

[6] Wei Feng, Fei Yin, Xu Yao Zhang, Wenhao He, and Cheng Lin Liu. Residual dual scale scene text spotting by fusing bottom-up and top-down processing. *International Journal of Computer Vision*, 129(3):619–637, 2021. 7

[7] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 369–376, 2006. 5

[8] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1742, 2006. 4

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 2, 3, 6

[10] Tong He, Weilin Huang, Yu Qiao, and Jian Yao. Text-attentional convolutional neural network for scene text detection. *IEEE Transactions on Image Processing (TIP)*, 25(6):2529–2541, 2016. 2

[11] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Multi-oriented and multi-lingual scene text detection with direct regression. *IEEE Transactions on Image Processing (TIP)*, 27(11):5406–5419, 2018. 2, 7

[12] Wenhao He, Xu Yao Zhang, Fei Yin, and Cheng Lin Liu. Deep direct regression for multi-oriented scene text detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 745–753, 2017. 2

[13] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. ICDAR 2015 competition on robust reading. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160, 2015. 7

[14] Vijeta Khare, Palaiahnakote Shivakumara, Raveendran Paramesran, and Michael Blumenstein. Arbitrarily-oriented multi-lingual text detection in video. *Multimedia Tools and Applications*, 76(15):16625–16655, 2017. 6

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. 6

[16] Deepak Kumar, M. N. Anil Prasad, and A. G. Ramakrishnan. Multi-script robust reading competition in ICDAR 2013. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 14:1–14:5, 2013. 2

[17] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *arXiv preprint arXiv:1908.08207*, 2019. 2, 4, 7

[18] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *AAAI Conference on Artificial Intelligence*, pages 4161–4167, 2017. 2

[19] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *AAAI Conference on Artificial Intelligence*, pages 11474–11481, 2020. 7

[20] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-Song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5909–5918, 2018. 7

[21] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision (ECCV)*, volume 8693, pages 740–755, 2014. 6

[22] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5676–5685, 2018. 4, 7

[23] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *European Conference on Computer Vision (ECCV)*, pages 19–35. Springer, 2018. 2

[24] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *European Conference on Computer Vision (ECCV)*, September 2018. 4, 7

[25] Rodrigo Minetto, Nicolas Thome, Matthieu Cord, Neucimar J. Leite, and Jorge Stolfi. Snoopertrack: Text detection and tracking for outdoor videos. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 505–508, 2011. 2, 7

[26] Wei-Yi Pei, Chun Yang, Li-Yu Meng, Jie-Bo Hou, Shu Tian, and Xu-Cheng Yin. Scene video text tracking with graph matching. *IEEE Access*, 6:19419–19426, 2018. 7

[27] Sangeeth Reddy, Minesh Mathew, Lluís Gómez, Marçal Rusiñol, Dimosthenis Karatzas, and C. V. Jawahar. Roadtext-1k: Text detection & recognition dataset for driving videos. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 11074–11080, 2020. 2, 7

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015. 3, 4

[29] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2550–2558, 2017. 7

[30] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NIPS)*, pages 802–810, 2015. 2

[31] Palaiahnakote Shivakumara, Liang Wu, Tong Lu, Chew Lim Tan, Michael Blumenstein, and Basavaraj S. Anami. Fractals based multi-oriented text detection system for recognition in mobile video images. *Pattern recognition (PR)*, 68:158–174, 2017. 6

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 5

[33] 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 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4651–4659, 2015. 2, 7

[34] Shu Tian, Wei-Yi Pei, Ze-Yu Zuo, and Xu-Cheng Yin. Scene text detection in video by learning locally and globally. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2647–2653, 2016. 1, 2

[35] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *European Conference on Computer Vision (ECCV)*, pages 56–72, 2016. 7

[36] Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, and Jiaya Jia. Learning shape-aware embedding for scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4234–4243, 2019. 7

[37] Lan Wang, Yang Wang, Susu Shan, and Feng Su. Scene text detection and tracking in video with background cues. In *Proceedings of the International Conference on Multimedia Retrieval (ICMR)*, pages 160–168, 2018. 2, 6, 7

[38] Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng. End-to-end text recognition with convolutional neural networks. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 3304–3308, 2012. 2, 4

[39] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9336–9345, 2019. 2

[40] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 8439–8448, 2019. 2

[41] Xiaobing Wang, Yingying Jiang, Shuli Yang, Xiangyu Zhu, Wei Li, Pei Fu, Hua Wang, and Zhenbo Luo. End-to-end scene text recognition in videos based on multi frame tracking. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 1255–1260, 2017. 1

[42] Yang Wang, Lan Wang, and Feng Su. A robust approach for scene text detection and tracking in video. *Advances in Multimedia Information Processing (PCM)*, 11166:303–314, 2018. 6, 7

[43] Liang Wu, Palaiahnakote Shivakumara, Tong Lu, and Chew Lim Tan. A new technique for multi-oriented scene text line detection and tracking in video. *IEEE Transactions on Multimedia (TMM)*, 17(8):1137–1152, 2015. 6

[44] Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R. Scott. Convolutional character networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9125–9135, 2019. 2, 7

[45] Chuhui Xue, Shijian Lu, and Fangneng Zhan. Accurate scene text detection through border semantics awareness and bootstrapping. In *European Conference on Computer Vision (ECCV)*, pages 355–372, 2018. 7

[46] Xue-Hang Yang, Wenhao He, Fei Yin, and Cheng-Lin Liu. A unified video text detection method with network flow. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 331–336, 2017. 1, 2, 7

[47] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1083–1090, 2012. 2

[48] Qixiang Ye and David S. Doermann. Text detection and recognition in imagery: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(7):1480–1500, 2015. 2

[49] Fei Yin, Yi-Chao Wu, Xu-Yao Zhang, and Cheng-Lin Liu. Scene text recognition with sliding convolutional character models. In *arXiv preprint arXiv:1709.01727*, 2017. 2, 4

[50] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(5):970–983, 2014. 6

[51] Xu-Cheng Yin, Ze-Yu Zuo, Shu Tian, and Cheng-Lin Liu. Text detection, tracking and recognition in video: A comprehensive survey. *IEEE Transactions on Image Processing (TIP)*, 25(6):2752–2773, 2016. 2

[52] Hongyuan Yu, Chengquan Zhang, Xuan Li, Junyu Han, Errui Ding, and Liang Wang. An end-to-end video text detector with online tracking. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 601–606, 2019. 1, 2, 4, 6, 7

[53] Xu Zhao, Kai-Hsiang Lin, Yun Fu, Yuxiao Hu, Yuncai Liu, and Thomas S. Huang. Text from corners: A novel approach to detect text and caption in videos. *IEEE Transactions on Image Processing (TIP)*, 20(3):790–799, 2011. 6

[54] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: An efficient and accurate scene text detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5551–5560, 2017. 2, 7

[55] Yingying Zhu, Cong Yao, and Xiang Bai. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, 10(1):19–36, 2016. 2

[56] Ze-Yu Zuo, Shu Tian, Wei-Yi Pei, and Xu-Cheng Yin. Multi-strategy tracking based text detection in scene videos. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 66–70, 2015. 1, 2, 7