Read while you drive - multilingual text tracking on the road

Sergi Garcia-Bordils[1,3], George Tom[2], Sangeeth Reddy[2], Minesh Mathew[2], Marçal Rusiñol[3], C.V. Jawahar[2], and Dimosthenis Karatzas[1]

[1] Computer Vision Center (CVC), UAB, Spain {sergi.garcia,dimos}@cvc.uab.cat
[2] Center for Visual Information Technology (CVIT), IIIT Hyderabad, India
{george.tom,sangeeth.battu,minesh.mathew,jawahar}@research.iiit.ac.in
[3] AllRead Machine Learning Technologies

Abstract. Visual data obtained during driving scenarios usually contain large amounts of text that conveys semantic information necessary to analyse the urban environment and is integral to the traffic control plan. Yet, research on autonomous driving or driver assistance systems typically ignores this information. To advance research in this direction, we present RoadText-3K, a large driving video dataset with fully annotated text. RoadText-3K is three times bigger than its predecessor and contains data from varied geographical locations, unconstrained driving conditions and multiple languages and scripts. We offer a comprehensive analysis of tracking by detection and detection by tracking methods exploring the limits of state-of-the-art text detection. Finally, we propose a new end-to-end trainable tracking model that yields state-of-the-art results on this challenging dataset. Our experiments demonstrate the complexity and variability of RoadText-3K and establish a new, realistic benchmark for scene text tracking in the wild.

Keywords: Scene text · tracking · multilingual · driving videos

1 Introduction

There is text in about 50% of the images in large-scale datasets such as MS Common Objects in Context [32], and the percentage goes up sharply in urban environments. Specific activities, such as making a purchase, using public transportation or finding a place in the city, are highly dependent on understanding textual information in the wild, and driving is a prime example.

Text on traffic signs is an integral part of the traffic control plan as it provides the driver with information on the upcoming situation. Nevertheless, textual information is currently not exploited by Advanced Driver Assistance Systems (ADAS) or autonomous driving systems. Automatic road text understanding could allow introducing new driving instructions in the route, updating maps automatically, and identifying target locations in the street. At the same time, much text on the road is a distraction for the driver. For example, 71% of Americans consciously look at billboard messages while driving [34] . As a matter

Fig. 1: Sample frames from the new RoadText-3K dataset taken from different locations containing multilingual text. The top-left frame was captured in the US (English text), the top-right frame in Spain (Spanish/Catalan) while bottom row frames are taken from India videos (Telugu and Hindi). Transcriptions are shown only for some of the bounding boxes to avoid clutter.

of fact, drivers who detected more traffic signs also detected more advertisements [31], as text naturally attracts bottom-up human attention [6].

The lifetime of text objects while driving is quite short. At normal city driving speeds (30 km/h), a road text instance enters (becomes readable) and exits the scene within 3-5 seconds. Thus a reading system for driving is required to detect, track and recognise text early on, at the initial instances of its occurrence, while the text is typically far from the vehicle. This requires a fast model, tolerant to occlusions, which can deal with tiny text instances, typically affected by motion blur and important perspective distortions, especially in the case of roadside text. While object tracking is a well explored area of research, there have only been a few attempts at extending these ideas to text tracking. Our dataset contains high-resolution videos where many of the text instances have a small size, suffer significant perspective changes during the sequence and present visual artifacts such as blurred or out of focus text. This makes the extension of object tracking to road-text tracking non-trivial.

In this work, we introduce a significant quantitative and qualitative extension to the RoadText-1k dataset [26] and a comprehensive study of baseline tracking methods before introducing a new tracking model that yields better performance at high speeds compared to the baselines.

Specifically our contributions are the following:

– We extend the existing RoadText-1K dataset by adding 2000 more videos. The extended RoadText-3K dataset contains videos captured from three

countries that contain text instances in three scripts — Latin (English / Spanish / Catalan), Telugu and Devanagari.
– We provide a detailed study of various tracking methods and compare their performance on RoadText-3K. We build multiple trackers based on state-of-the-art scene text detectors and highlight the key aspects that influence tracking in each of the approaches.
– We propose a new tracking approach using a CenterNet [37] based text detector. Our approach outperforms other trackers in terms of MOTA and MOTP metrics while maintaining real-time speed.

2 Related Work

2.1 Datasets for text spotting in videos

Existing datasets for spotting text in videos are ICDAR Text in Videos dataset [15], YouTube Video Text (YVT) [24], RoadText-1K [26], Large-scale Video Text dataset (LSVTD) [7] and Bilingual Open World Video Text (BOVText) [35].

ICDAR Text in Videos dataset was introduced as part of 2013–2015 Robust Reading Challenge in ICDAR. It contains 51 videos (28k frames) of varying lengths, captured in different scenarios such as highways, shopping in a supermarket or walking inside buildings. The videos are captured using a handheld device or a head-mounted camera. YVT contains 30 videos (13k frames) in total, sourced from Youtube. The videos contain scene text and born digital overlay text such as captions, titles or logos. RoadText-1K [26] has $1,000$ driving videos (300k frames) with annotations for text detection, recognition and tracking. The videos contain only text in English since all the videos are captured from the United States. LSVTD [7] has 100 videos (65k frames) captured in 13 indoor and 8 outdoor scenarios. BoVText is a recently introduced dataset with 2021 videos (1,750k frames). The dataset contains both born digital overlay text and scene text instances. The videos are harvested from video-sharing platforms, and consequently, there are videos captured from different parts of the world.

The newly introduced RoadText-3K contains driving or road videos and it is an extension of the existing RoadText-1K. 2000 new videos from two different geographical locations are added to the existing RoadText-1K to make the new RoadText-3K dataset. Among the existing datasets, BoVText, a work that is concurrent to ours is the only dataset that has more number of videos and text instances in it compared to RoadText-3K. Compared to BOVText, which has text instances in Chinese and English, RoadText-3k has annotated text instances in Latin (English, Spanish and Catalan), Telugu and Devanagari (Hindi).

2.2 Text Detection

Text detection approaches can be classified into two types — regression-based and segmentation-based. TextBoxes++ [18] and CTPN [30] are examples for regression-based methods. TextBoxes++ generates proposals using a quadrilateral representation of the bounding boxes. CTPN uses vertical anchors of fixed

width to predict the location of text. The model combines the output of a Convolutional Neural Network (CNN) with a Recurrent Neural Network (RNN) to build more meaningful text proposals. Segmentation based methods include EAST [38], which generates dense pixel-level proposals. It employs non-maximal suppression to filter the proposals. FOTS [20] generates pixel-level predictions, outputting a confidence score, the distance, and the rotation of the bounding box the pixel belongs to. Models such as CRAFT [2] focus on detecting curved text. CRAFT uses a more unconventional bottom-up approach and learns to output individual character predictions and their affinity (whether they belong to the same word or not). Since most datasets do not include character-level annotations, CRAFT uses a weakly-supervised method to generate ground truth.

2.3 Text Tracking

Text tracking methods cover both main families of tracking approaches — Tracking by Detection (TbD) and Detection by Tracking (DbT). In TbD, text instances in each individual frame are detected. Then, subsequent detections that correspond to the same text instance are linked to form a track [7, 25, 33]. For example, [7, 25] use spatio-temporal redundancy between frames to track text instances across frames.

In the case of DbT, text detection is performed in an initial frame and these detections are then propagated to the subsequent frames using a propagation algorithm. Detection is then repeated at set intervals to update the trackers. In [10] text regions are extracted using a Maximally Stable External Regions (MSER)-based detector [9] every 5 frames, and are then propagated for the next 5 frames using MSER propagation. Snooper-track [22] uses a similar strategy, where text is tracked using a particle filter system. In [28] a combination of TbD and DbT is used (spatio-temporal learning and template matching) to improve text tracking. The authors use the Hungarian algorithm to do the final association of the detections. In [29] TbD and DbT are explored for the problem of tracking and recognizing embedded captions in online videos.

In addition to the above two categories of approaches, there have been some end-to-end approaches that do detection and tracking simultaneously. In [36] a Convolutional Long Short Term Memory (ConvLSTM) is used in the detection branch to capture spatial structure information and motion memory. Yu et al. [36] proposed an end-to-end tracking model where a two branch network is used to detect and track text instances simultaneously.

2.4 Multiple Object Tracking Metrics

Evaluation of multiple object tracking has evolved extensively over the past years [3, 21]. In this work, we evaluate the tracking of text instances using four standard metrics — MOTA, MOTP, IDF1 and IDs. Multiple Object Tracking Accuracy(MOTA) takes into account false positives, false negatives and ID switches at track level. MOTP (Multiple Object Tracking Precision) measures the similarity between the true positive detections and their corresponding

ground truth objects (in this case, the similarity is measured in terms of the Intersection over Union). IDF1 is similar to the F-score metric used in binary classification, it reports the harmonic mean of identification precision and recall. ID switches (IDs) indicates the number of re-identifications of a tracked object.

3 RoadText-3K Dataset

We introduce the RoadText-3K dataset, an extension of the existing RoadText-1K [26] dataset. We extend the former by adding new 2000 videos captured from two different geographical locations and containing text in 6 languages, including English. RoadText-1K has videos captured from the United States. In the new 2000 videos, 1000 are captured from Spain, and the remaining 1000 are from India. The dataset can be downloaded from https://datasets.cvc.uab.es/roadtext3k/.

Table 1: RoadText-3K is an extension of RoadText-1K. The new dataset has 2000 more videos that are captured in locations in two continents making it ideal for detection and tracking of multilingual text on roads.

Dataset	RoadText-1K [26]	RoadText-3k
Source	car-mounted	car-mounted
Videos	1,000	3,000 (2,000 new videos)
Length (seconds)	10	10
Resolution	1280 x 720	1280 x 720
Annotated frames	300,000	927,974
Text Instances	1,280,613	4,039,250
Tracks	28,280	88,427
Unique words	8,263	22,115
Location	US	US, Europe and India
Scripts	Latin	Latin, Telugu and Devanagari

Similar to RoadText-1K, the new videos are annotated with bounding boxes of text tokens in each frame, transcriptions for the text tokens and track information. Videos from each of the three locations are split in 50:20:30 ratio to train, validation and test splits respectively.

3.1 Videos

Videos are collected from various geographical locations to accommodate the diversity of scripts scenes and geographies. Out of the 2,000 new videos, 1,000 videos are collected from India and the other 1,000 are collected from Europe. Videos are captured with a camera mounted on a vehicle.

3.2 Annotations

We follow the same annotation approach as RoadText-1K [26]. Annotations include text bounding box, text transcription and track id. Text bounding boxes

are added at line level as in RoadText-1K [26]. The videos from Europe have text in Spanish, Catalan and English. Videos captured in India include text in English, Telugu and Hindi. Few text instances do not belong to any of these languages and are labelled as "Others". Transcriptions are not provided for text instances in this category.

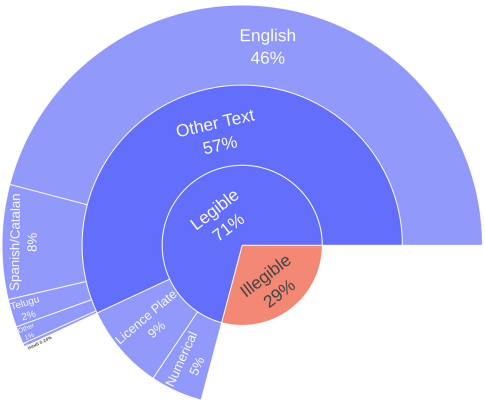

Fig. 2: Distribution of text instances in Roadtext-3K based on legibility, type and language.

3.3 Analysis

Basic statistics of the dataset in comparison with Roadtext-1K are shown in Table 1. The distribution of text instances based on their legibility, text type and language is shown in Figure 2. Around 29% of the text instances are illegible. This is expected since texts in driving videos are subject to various artifacts including low resolution, motion blur, glare and perspective distortions. Indian roads and highways are dominated by English text, leading to a low percentage of Indian language text instances. The distribution of track lengths in the dataset is shown in Fig 3. The lifetime of text instances in the driving videos is generally short, with most tracks having a duration of < 1 second (< 30fps). Fig 4 shows a word cloud of the most common text tokens in the dataset. It can be seen that tokens like "P" and "30" are among these, suggesting that there are many text tokens from traffic/road boards.

4 Methodology

We first evaluate state-of-the-art scene text detection models on individual frame of the videos in RoadText-3K to identify efficient detectors for our tracking methods. For text tracking, multiple baseline trackers are built using these text detectors in both TbD and DbT paradigms. Finally, we propose a new TbD approach that uses CenterNet for detections.

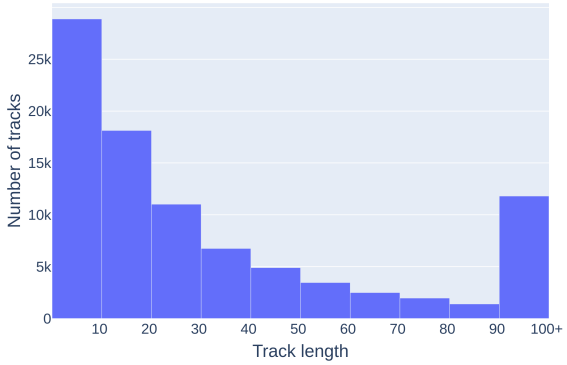

Fig. 3: Distribution of track lengths in RoadText-3K.

4.1 Text Detection

To test the performance of modern text detectors on RoadText-3K, we have evaluated CTPN [30], EAST [38], FOTS [20] and CRAFT [2] on the test split of our dataset. Since consecutive frames contain similar text instances, we evaluate these detectors on every 10th frame in a video. All four detection models were originally trained to detect single words. Since text instances in our dataset are annotated at line level, we have fine-tuned these models on the train split of the RoadText-3K. For CTPN, EAST and FOTS, we have used implementations available online, which include training and evaluation code[4]. The authors of CRAFT provide an implementation of their model but do not include the training code, so we have used their provided pre-trained model on SynthText [11], ICDAR 2013 [16] and ICDAR 2015 [15] datasets. Since we have many small text instances, each frame is resized to an input size of 1280x720.

4.2 Text Tracking

Both TbD and DbT trackers are built using the above mentioned text detection models. In addition, we propose a new TbD model that uses CenterNet for temporally aware text detection. We quantitatively evaluate our methods using the previously introduced MOT metrics[5], which have also been used in the ICDAR 2013 [16] and ICDAR 2015 [15] challenges.

[4] For CTPN, EAST and FOTS we have used unofficial implementations of the original methods, for CRAFT we have used the author's released implementation:

- CTPN: https://github.com/eragonruan/text-detection-ctpn
- EAST: https://github.com/argman/EAST
- FOTS: https://github.com/jiangxiluning/FOTS.PyTorch
- CRAFT: https://github.com/clovaai/CRAFT-pytorch

[5] We used the implementation given in https://github.com/cheind/py-motmetrics

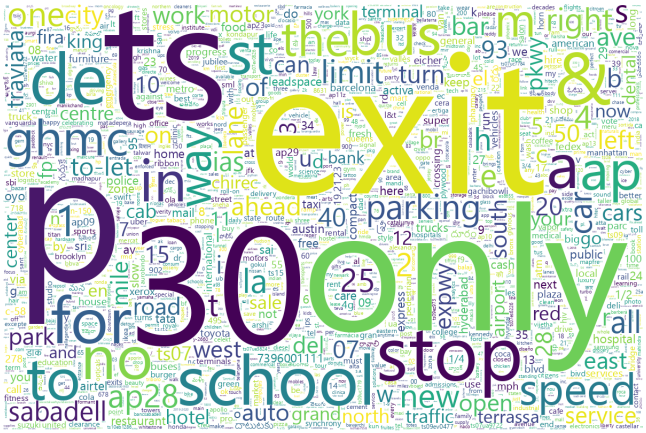

Fig. 4: WordCloud of the text tokens in the dataset. The most common text comes from road signs(for example, “P”, “stop”, “30” and “exit”) and the common prefixes on Indian licence plates (for example ”ts” and “ap”).

Tracking By Detection (TbD) In the TbD paradigm, trackers seek to associate detections between frames using temporal and visual information. These methods perform text detection on every frame, which can make the text tracking system slower. Since detections are made independently on each frame, flickering, detection merging might occur, while the final tracking result is affected by any detection failures. In order to evaluate TbD on our dataset, we use text detection from the scene text detectors discussed in the previous section and use SORT [4] to associate the instances across frames. SORT uses Kalman filters to predict the position of the objects (text instances in our case) in consecutive frames. Using the IoU as the distance, it solves the matching problem between predicted positions and frame detections using the Hungarian algorithm [17].

Detection By Tracking (DbT) In DbT every single object is explicitly tracked by a different instance of the tracker. This technique can employ visual and temporal information to find the location of the object in consecutive frames, while keeping the inference times down. One of the major drawbacks is that, in the case of multiple object tracking, we need to initialize a new instance of the tracker for every new object. We have opted for a setup similar to the ones proposed in [10, 22]. Every 5 frames, we perform text detection using one of the text detection models that we discuss in section 4.1. For every new text instance found, we launch a tracker that will follow the text instance for the next 5 consecutive frames. In the final, frame we compare the location of the tracked objects with the detections of the text detector on the current frame. When the IoU between a tracked instance and one of the detections surpasses 0.5, we consider this a match, and the tracker continues following the text instance for the next 5 frames. When no tracked instances match any of the detections, we

launch a new tracker. Finally, if no detection matches a tracked instance, we keep tracking it for 5 more frames, but if no detections match it 5 frames later the tracker stops. This avoids relying too much on the detections of the detector, since motion blur or temporal occlusion can introduce detection failures. This also introduces the risk of increasing the number of false positives, since a text instance that leaves the scene or gets permanently occluded may still get tracked for a few frames.

FOTS obtained the highest F-score and recall from all the detection methods, as well as a relative high inference speed. For this reason, we have used its detections to start and stop the trackers. To perform the tracking of the text instances we use CSR-DCF [1], KCF [13] and MedianFlow [14]. These trackers use traditional approaches to object tracking such as correlation filters and kernels.

4.3 CenterNet-based detection and tracking

Our dataset features high resolution images with many small text instances. While downsampling frames allows faster inference, the detection of small instances requires working at a higher resolution. Nevertheless, the complexity of many modern text detectors results in slow inference speeds on high-resolution images. We have tried to simplify the approach towards scene text detection and tracking on RoadText-3k, and we have tailored a framework that focuses on real-time inference on high-resolution inputs. Our approach is based on CenterNet [37], and we use it for both text detection and tracking. We made the code and the weights publicly available at https://github.com/Sergigb/roadtext3k-baselines.

CenterNet is an object detection model that represents objects as a single point at their bounding box center, and then regresses the width and height at the center of the location. The centers of the objects are represented as a Gaussian kernel on a heatmap and focal loss [19] is used to learn this representation. To achieve better performance, we adopt ResNet-18 [12] as a backbone to our networks, which offers a good balance between performance and real-time inference. Inspired by YOLOv4 [5], we replace ReLU with the Mish activation function [23]. We evaluated our approach on single frame object detection in the same fashion as the previous object detectors.

For our tracking model, we have tried to leverage temporal information into our CenterNet-based text detector. We add temporal awareness to the model by adding one convolutional GRU cell before upsampling the feature map. This cell which is similar to the convolutional LSTM model presented in [27], but uses the GRU cell [8] layout. In the decoder of the network we apply transpose convolutions to upscale the latent feature map. Similar to the other TbD approaches we discuss in 4.2, we use SORT to perform the object association between the frames. Figure 5 shows an overview of the architecture.

5 Results

Results of the detection and tracking experiments are presented in this section. All experiments results are reported on the test split of the Roadtext-3K dataset.

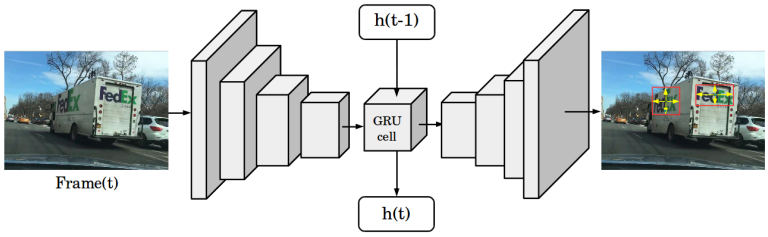

Fig. 5: Our method extends the CenterNet architecture with a convolutional GRU cell at the output of a ResNet-18 to aggregate spatial and temporal information. $h(t)$ represents the hidden state of the GRU cell in the frame t.

5.1 Frame level text detection

We consider a detection to be a true positive if it overlaps with any ground truth instance with an IoU of over 0.5. Table 2 shows results of the frame level text detection. FOTS obtains the best F-score and the highest recall. Our CenterNet-based method gets competitive performance while being the fastest method we have tested, obtaining an inference speed of 44 FPS. Despite not being fine-tuned on our dataset, CRAFT obtains the highest precision but has the lowest FPS.

Table 2: Results of frame level text detection. FOTS has the highest F-score and recall, while our method obtains the fastest inference speed.

Detector	Precision(%)	Recall(%)	F-score(%)	FPS
CTPN	34.62	32.74	33.65	13
EAST	32.14	29.51	31.27	17
FOTS	42.77	**50.74**	**46.41**	19
CRAFT	**54.3**	37.21	44.15	5
CenterNet	50.3	39.1	43.9	**44**

5.2 Tracking

TbD: Results of the TbD on the test split of RoadText-3K are shown in Table 3. Our CenterNet+GRU model and CRAFT obtain similar scores. Both methods have high precision and similar recall, and both models produce a low number of ID switches. Using GRU with CenterNet reduces the number of ID switches (column IDs) by a large margin and improves the MOTA score. We hypothesize that the GRU cell helps reducing the flickering and improves the consistency of the tracking. Speed is reported based on the combined time taken to run the detector and associate detections using SORT. Our CenterNet-based approaches have the highest inference speeds, reaching 40 FPS when we do not use the GRU cell and 31 when we use it.

Table 3: Results of Tracking by Detection (TbD) using different detectors. In all cases the SORT algorithm [4] is used for associating detections in two consecutive frames.

Detector	MOTA(%)	IDF1(%)	Recall(%)	Precision(%)	IDs	MOTP(%)	FPS
CTPN	24.93	51.60	52.00	66.80	15524	65.11	11
EAST	25.20	51.33	49.20	68.00	13178	69.47	12
FOTS	28.47	**56.57**	**59.73**	66.33	12883	71.82	14
CRAFT	35.40	54.77	46.20	**81.73**	**6328**	70.59	5
CenterNet	33.80	54.80	53.00	74.50	15032	72.44	**40**
CenterNet+GRU	**36.00**	54.80	47.60	81.30	8896	**72.74**	31

DbT: Table 4 shows results of the three different DbT methods we evaluated. It can be seen that precision is much lower for all the three approaches compared to TbD results shown in Table 3. Lower precision is partly due to a higher number of false positives. One possible reason for this is the fact that even text instances that have disappeared are tracked until the next set of detections are available. However recall scores for DbT approaches are comparable to that of TbD. For example CSR-DCF has a recall of 54.10%, second only to FOTS in the TbD setup. DbT is usually a good choice if inference speed is a constraint. Note nevertheless that the proposed CenterNet-based TbD approach still yields competitive speeds (35 FPS).

Table 4: Results of detection by tracking (DbT) approach using various trackers. FOTS [20] is used for detections in all cases.

Tracker	MOTA(%)	IDF1(%)	Recall(%)	Precision(%)	IDs	MOTP(%)	FPS
CSR-DCF	**-33.70**	**35.30**	**54.10**	**38.70**	**16136**	71.66	25
KCF	-51.80	31.10	49.60	33.40	19534	71.8	76
MEDIANFLOW	-53.00	28.10	48.40	32.90	22614	**71.81**	**83**

5.3 Qualitative Analysis

Visually inspecting the results of the different methods gives us a hint of how they behave under different conditions. For example, the two approaches we tested behave very differently in cases of temporal occlusions. We can see one of such cases in Fig 6, where the vehicle's windscreen wiper temporally occludes the text being tracked. Since the detector does not have any sort of temporal awareness, in TbD the occluded text fails to be localized. In this case, SORT managed to recover from the occlusion and correctly reassigned the IDs in the following frames. Our model displays a similar behaviour to the TbD approach, but SORT recovers again from the occlusions. In DbT, the CSR-DCF algorithm manages to keep tracking the text instances even under partial occlusion. This robustness to temporal occlusions minimizes the probabilities of ID switches. However, the two text instances in the upper left side of the board ("SOUTH" and "678") appear

to have slightly shifted bounding boxes after the occlusion. Since we check the detections every 5 frames to start and stop trackers, a shifted bounding box can result in ID switches if it does not match any detection. This could be attributed to the fact that CSR-DCF uses the last known visual appearance of the object to find it in the next frame. Even though TbD generally offers more precise and reliable detection overall, distant or blurry text can introduce flickering. As seen in Fig 7a, the two lower text instances disappear for a few frames and then reappear, increasing the chances of ID switches. In DbT (Fig 7b), the proposals keep being reliably tracked. Our model displays a more conservative behaviour, one of the smaller instances is not tracked but the others are consistently tracked.

(a) TbD using FOTS and SORT

(b) DbT using FOTS to start and stop CSR-DCF trackers

(c) Our CenterNet+GRU approach.

Fig. 6: Results of TbD, DbT, and our CenterNet-based model in case of a temporal occlusion. Numbers shown alongside the boxes are track numbers.

Judging by the quantitative results, the TbD approach seems to obtain better overall tracking results (best MOTA, lowest ID switches, etc.), but qualitative results suggest that DbT can be advantageous in cases with occlusions or detector failures. The biggest drawback of DbT is the increase in false positives, and this is primarily due to the inability to immediately stop tracking when a text instance leaves the scene. The lower recall in DbT (5.64% between FOTS + CSR-DCF and FOTS + SORT) can be partially explained by the fact that we check for new detections every 5 frames, which can delay starting a new tracker and increase the amount of false negatives.

6 Conclusions

We have introduced RoadText-3K, an extension to the existing RoadText-1K dataset with an additional 2000 driving videos captured in different geographical locations and containing text in different scripts and languages. We evaluated

(a) TbD using FOTS and SORT

(b) DbT using FOTS to start and stop CSR-DCF trackers.

(c) Our CenterNet+GRU method.

Fig. 7: Performance of the various models under a case of detection flickering. Numbers shown along with the boxes are track numbers.

several state-of-the-art detectors in this dataset and employed them to construct tracking by detection and detection by tracking methods. Results demonstrate that driving videos are especially challenging. Finally, we have presented a new simple and efficient approach for tracking by detection which incorporates temporal information in the detection branch. Our method yields competitive tracking results while obtaining real-time inference speeds.

7 Acknowledgements

This work has been supported by the Pla de Doctorats Industrials de la Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya; Grant PDC2021-121512-I00 funded by MCIN /AEI/ 10.13039/501100011033 and the European Union NextGenerationEU/PRTR; Project PID2020-116298GB-I00 funded by MCIN/ AEI /10.13039/501100011033; Grant PLEC2021-007850 funded by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR; Spanish Project NEOTEC SNEO-20211172 from CDTI and CREATEC-CV IMCBTA/2020/46 from IVACE and IHub-Data at IIIT-Hyderabad.

References

1. Lukežič Alan, Tomáš Vojíř, Luka Čehovin, Jiří Matas, and Matej Kristan. Discriminative correlation filter tracker with channel and spatial reliability. *IJCV*, 126, 2018.
2. Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *CVPR*, 2019.
3. Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008.
4. Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016.
5. Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
6. Moran Cerf, E Paxon Frady, and Christof Koch. Faces and text attract gaze independent of the task: Experimental data and computer model. *Journal of vision*, 9, 2009.
7. Zhanzhan Cheng, Jing Lu, Baorui Zou, Liang Qiao, Yunlu Xu, Shiliang Pu, Yi Niu, Fei Wu, and Shuigeng Zhou. Free: A fast and robust end-to-end video text spotter. *IEEE Transactions on Image Processing*, 30:822–837, 2020.
8. Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
9. Michael Donoser and Horst Bischof. Efficient maximally stable extremal region (mser) tracking. In *CVPR*, 2006.
10. Lluis Gomez and Dimosthenis Karatzas. Mser-based real-time text detection and tracking. In *ICPR*, 2014.
11. Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, 2016.
12. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
13. Joao F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.
14. Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In *ICPR*, 2010.
15. Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *ICDAR*, 2015.
16. Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazàn Almazàn, and Lluís Pere de las Heras. ICDAR 2013 Robust Reading Competition. In *ICDAR*, 2013.
17. H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2, 1955.
18. Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *TIP*, 27, 2018.

19. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.
20. Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. FOTS: Fast oriented text spotting with a unified network. In *CVPR*, 2018.
21. Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.
22. Rodrigo Minetto, Nicolas Thome, Matthieu Cord, Neucimar J Leite, and Jorge Stolfi. Snoopertrack: Text detection and tracking for outdoor videos. In *ICIP*, 2011.
23. Diganta Misra. Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*, 2019.
24. Phuc Xuan Nguyen, Kai Wang, and Serge Belongie. Video text detection and recognition: Dataset and benchmark. In *WACV*, 2014.
25. Marc Petter, Victor Fragoso, Matthew Turk, and Charles Baur. Automatic text detection for mobile augmented reality translation. In *ICCV Workshops*, 2011.
26. Sangeeth Reddy, Minesh Mathew, Lluis Gomez, Marçal Rusinol, Dimosthenis Karatzas, and CV Jawahar. Roadtext-1k: Text detection & recognition dataset for driving videos. In *ICRA*, 2020.
27. 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. *NeurIPS*, 2015.
28. Shu Tian, Wei-Yi Pei, Ze-Yu Zuo, and Xu-Cheng Yin. Scene text detection in video by learning locally and globally. In *IJCAI*, 2016.
29. Shu Tian, Xu-Cheng Yin, Ya Su, and Hong-Wei Hao. A unified framework for tracking based text detection and recognition from web videos. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):542–554, 2017.
30. Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, 2016.
31. Darja Topolšek, Igor Areh, and Tina Cvahte. Examination of driver detection of roadside traffic signs and advertisements using eye tracking. *Transportation research part F: traffic psychology and behaviour*, 43, 2016.
32. Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. In *arXiv preprint arXiv:1601.07140*, 2016.
33. Xiaobing Wang, Yingying Jiang, Shuli Yang, Xiangyu Zhu, Wei Li, Pei Fu, Hua Wang, and Zhenbo Luo. End-to-end scene text recognition in videos based on multi frame tracking. In *ICDAR*, 2017.
34. Diane Williams. The arbitron national in-car study. *Arbitron Inc*, 2009.
35. Weijia Wu, Debing Zhang, Yuanqiang Cai, Sibo Wang, Jiahong Li, Zhuang Li, Yejun Tang, and Hong Zhou. A bilingual, openworld video text dataset and end-to-end video text spotter with transformer. *NeurIPS 2021 Track on Datasets and Benchmarks*, 2021.
36. Hongyuan Yu, Yan Huang, Lihong Pi, Chengquan Zhang, Xuan Li, and Liang Wang. End-to-end video text detection with online tracking. *PR*, 2021.
37. Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.
38. 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.