State-of-the-Art in Action: Unconstrained Text Detection

Diep Thi Ngoc Nguyen
University of Engineering and Technology, VNU, Hanoi
144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
ngocdiep@vnu.edu.vn
HBLab Jsc. chupi@hblab.vn

Abstract

In this paper, we stage five real-world scenarios for six state-of-the-art text detection methods in order to evaluate how competent they are with new data without any training process. Moreover, this paper analyzes the architecture design of those methods to reveal the influence of pipeline choices on the detection quality. The setup of experimental studies are straight-forward: we collect and manually annotate test data, we reimplement the pretrained models of the state-of-the-art methods, then we evaluate and analyze how well each method achieve in each of our collected datasets. We found that most of the state-of-the-art methods are competent at detecting textual information in unseen data, however, some are more readily used for real-world applications. Surprisingly, we also found that the choice of a post-processing algorithm correlates strongly with the performance of the corresponding method. We expect this paper would serve as a reference for researchers as well as application developers in the field.

All collected data with ground truth annotation and their detected results is publicly available at our Github repository: https://github.com/chupibk/HBlab-rlq19.

1. Introduction

Technology development benefits tremendously from new research achievements. Adapting new state-of-the-art (SOTA) methods is always desirable with the hope to obtain the best or better performance. However, research methods are studied on carefully curated datasets, their achievement may not remain once the domains of data are shifted. In the text detection and recognition field, the same happens. To become a state-of-the-art method in this field is to achieve better scores over time in existing benchmark datasets such as ICDAR 13 Scene Text [5], ICDAR 15 Incidental Text [4], SVT [11], SCUT-CTW1500 curved text [15], and many others. On the other hand, real-world target data are often less complicated and applications are more about the robustness and quality of the detected results. Moreover, the resources to collect a sufficient data amount to train a deep learning model from scratch are often scarce.

This paper investigates how readily competent a model with pretrained weights of a SOTA method can be when applying to different text detection problems. We stage five common real-world scenarios for experiments as follows:

1. Digitalization of purchase receipts for accounting purpose;
2. Extraction of photo captions of pictures (e.g., pictures in museums, exhibitions) for information indexing;
3. Payment card reading for financial services;
4. Business name cards organizing;
5. Indexing product names and prices in supermarkets.

We collect objects of each category, capture their photos using smartphone cameras then manually annotate the captured image data. We select and reimplement six SOTA methods. Then we conduct both qualitative and quantitative experiments to evaluate how well each method obtain on each of our collected datasets. The analysis of the architecture design of methods are bottom-up. We use the detected results of each method to understand how each pipeline design affects its overall performance.

The paper is organized in six sections. In Section 2 we describe how we collect and annotate five datasets. Section 3 describes six state-of-the-art methods that we select and why we select them for our experiments. Section 4 and Section 5 are the main sections that show the experimental results and give discussion on what can be learned from them. Section 6 summarizes and gives a list of future works.

For audiences with engineering purposes, it is suggested to look at Section 4 and Section 5 without any loss of important information. For more eager audiences, one eight-page paper is not too long to follow.

2. Collecting and annotating test data

Five datasets which correspond to five real-world scenarios in this paper are: "BILL", "CAPTION", "CARD", "NAME", and "PRODUCT". For each category, we collect some sample objects, capture their photos using smartphones, and then assign one annotator to make the ground truth bounding boxes for each image. There are different objects to be captured in CAPTION and PRODUCT. In BILL and NAME, same objects can be captured twice or more. The CARD datasets contains photos of only one object but in many different orientations, light conditions, and backgrounds. Overall, the datasets vary from their categories, languages of textual information, and the conditions when images are captured. Each dataset has 20 or 21 images. The overview of five datasets are summarized in Table 1.

Ground truth is annotated for each image and comprises the bounding quadrilateral of each text line. No transcription of the words are provided since we focus only on text detection problem. It is to note that we do not annotate "do not care" text boxes. It is because when deploying a text detection method in real-world application, it is unusual to know before hand which texts belongs to "do not care" without any extra processing such as recognition or classification. The annotator only annotates the bounding boxes for the text lines in an image that she thinks to be important for the application scenarios. Figure 1 shows some sample images and their ground truth bounding boxes from our collected datasets.

The number of bounding boxes is different in each image. However, the number of bounding boxes in images of the same datasets is approximately the same. The average number of bounding boxes in BILL, CAPTION, CARD, NAME, PRODUCT are 39, 4, 16, 12, and 7, respectively.

3. Selecting State-of-the-art methods

3.1. Six state-of-the-art methods

We chose six state-of-the-art methods which are CRAFT [1], CTPN [9], EAST [16], FOTS [7], PixelLink [2], PSENet [12]. Table 2 shows a brief summary of why we selected those methods. We mainly look for methods which are at the top on the leaderboards of Robust Reading Competition or widely appraised by the community which is indicated by the number of stars or forks on Github. We assume that standing-at-the-top is state-of-the-art and widely-used means the methods are stable in many applications. We don't assume these are better than non-chosen methods.

3.2. Implementation of SOTA methods

We obtained the available source code of SOTA methods and installed them on our local machine. The machine runs on Ubuntu 16.4.5, with Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, 16GB RAM and a 2GB GeForce GTX 1080 Ti GPU.

The list of the links to source codes is as following:

- CRAFT [1]: pytorch, https://github.com/clovaai/CRAFT-pytorch
- CTPN [9]: keras, https://github.com/eragonruan/text-detection-ctpn
- EAST [16]: tensorflow, https://github.com/argman/EAST
- FOTS [7]: pytorch, https://github.com/Vipermdl/OCR_detection_IC15
- PixelLink [2]: keras, https://github.com/opconty/pixellink_keras
- PSENet [12]: pytorch, https://github.com/whai362/PSENet

Note that we did not use the official source code of FOTS at https://github.com/xieyufei1993/FOTS because the pretrained weights are not available. Likewise, we could not reproduce the official source code of PixelLink at https://github.com/ZJULearning/pixel_link. We instead used their reimplementation of their codes for experiments.

4. State-of-the-Art in action

4.1. Evaluation protocol: W2LEval

We will evaluate a SOTA method based on the speed (processing time) and the detection quality (precision and recall).

The evaluation of a detection as a match or a correct detection is commonly done by a threshold-based protocol such as the *Intersection over Union* (IoU) protocol [10] or the DetEval protocol [13]. Both are also used in Robust Reading Competition.

Let G be a collection of N ground truth text areas G_i and D be a collection of M detected text areas D_j. The IoU protocol defines a match if the detection D_j overlaps a ground truth G_i by more than a threshold s as:

$$\text{IoU}(G_i, D_j) = \frac{D_j \cap G_i}{D_j \cup G_i} \geq s \tag{1}$$

where $s = 0.5$ is commonly used.

However, if the ground truth is annotated in text lines but the detection results are in text words, the $\text{IoU}(G_i, D_j)$ will be very small and there will be not many matches. The DetEval protocol compensates for these cases by considering not only one-to-one matches but also one-to-many

Dataset	Image count	Description	Language	Image status
BILL	20	Purchase receipts (e.g., at supermarkets, restaurants, convenient stores, post offices)	Japanese, Vietnamese	varying angles, handwriting included, distorted
CAPTION	20	Pictures with text captions	Vietnamese, English	Varying angles, slightly blurred, incidental text
CARD	21	A prepaid visa card (member card)	Japanese, English	varying angles, may out of focus, complex background
NAME	21	Business name cards	Japanese, English, Vietnamese	varying angles, format and fonts; old name card included; horizontal and vertical text
PRODUCT	20	Product labels of goods at a supermarket	Vietnamese	varying angles, handwriting included, incidental text

Table 1: Overview of our five collected datasets for experiments

Figure 1: Sample images with their annotated bounding boxes from our collected datasets: BILL, CAPTION, CARD, NAME, PRODUCT

Method	Published year/venue	Reason
CRAFT [1]	CVPR 2019	Rank #1 at Robust Reading Competition, Focused Scene Text 2013-2015[1]
CTPN [9]	ECCV 2016	1st starred repository on Github (2,226)[2]
EAST [16]	CVPR 2017	2nd starred repository on Github (1,847)[3]
FOTS [7]	CVPR 2018	Rank #1 at the state-of-the-art leaderboards (IC15 dataset)[4]
PixelLink [2]	AAAI 2018	Highly starred repository on Github; #2 method with source code available at Ranking table of Robust Reading Competition, Incidental Scene Text 2015[5]
PSENet [12]	CVPR 2019	Rank #3 in Curved Text Detection on SCUT-CTW1500 leaderboard; highly starred repository on Github; #1 method with source code available at Ranking table of Robust Reading Competition, Incidental Scene Text 2015[5]

[1] https://rrc.cvc.uab.es/?ch=2&com=evaluation&task=1
[2] https://github.com/eragonruan/text-detection-ctpn
[3] https://github.com/argman/EAST
[4] https://paperswithcode.com/task/scene-text-detection
[5] https://rrc.cvc.uab.es/?ch=4&com=evaluation&task=1&e=1&f=1&d=0&p=0&s=1

Table 2: Selected state-of-the-art methods in text detection (as of July 29, 2019)

(splits) and many-to-one (merges) matches. The one-to-many type is when a ground truth is matched by a group of detection (i.e., the ground truth is splitted in to many areas). The many-to-one type is when a group of ground truth is matched by a detection (i.e., the ground truth is merged into one area). The many-to-many type is not suppported since the experimental observation shows this type is very infrequent.

The DetEval protocol [13] uses two ovelapping matrices σ and τ as originally proposed by Liang *et al*. [6] where σ relates to the area recall and τ relates to the area precision:

$$\sigma_{ij} = \mathrm{R}_{AR}(G_i, D_j) = \frac{Area(G_i \cap D_j)}{Area(G_i)} \tag{2}$$

$$\tau_{ij} = \mathrm{P}_{AR}(G_i, D_j) = \frac{Area(G_i \cap D_j)}{Area(D_j)} \tag{3}$$

A match in the DetEval protocol is defined using two thresholds for area recall and precision $t_r \in [0, 1]$ and $t_p \in [0, 1]$. A one-to-one match of D_j to G_i is if $\sigma_{ij} > t_r$ and $\tau_{ij} > t_p$.

A one-to-many match of a group S_k of D_j to G_i is if each detection D_j overlaps enough with the ground truth and a sufficient large proportion of the ground truth G_i has been detected:

$$\forall D_j \in S_k : \tau_{ij} \geq t_p \text{ and } \sum_{D_j \in S_k} \sigma_{ij} \geq t_r \tag{4}$$

A many-to-one match of a detection D_j to a group S_m of G_i is when each ground truth has been detected with a sufficient area precision and a large enough proportion of each ground truth has been detected:

$$\sum_{i \in S_m} \tau_{ij} \geq t_p \text{ and } \forall G_i \in S_m : \sigma_{ij} \geq t_r \tag{5}$$

Since our collected data are annotated in line-level whereas most of the SOTA methods predict word-level text areas, we choose the DetEval protocol without many-to-one type. Additionally, in Equation 4, we use a cascaded union of the group S_k of D_j instead of theirs $\sum \sigma_{ij}$. A cascaded union is more suitable for evaluating whether a group of detection covers a large portion of the ground truth because there is a situation that $D_j \in S_k$ overlaps much with each other but do not actually cover a large portion of the ground truth if using their arithmetic sum. We call this modified version of the DetEval protocol as W2LEval protocol.

Finally, the precision and recall scores for one image in W2LEval are defined as follows:

$$\mathrm{P} = \frac{\sum_j Match_D(D_j, G, t_r, t_p)}{|D|} \tag{6}$$

$$\mathrm{R} = \frac{\sum_i Match_G(G_i, D, t_r, t_p)}{|G|} \tag{7}$$

where $Match_D = 1$ if D_j is a one-to-one or one-to-many match to a ground truth; and $Match_G = 1$ if G_i is detected by a one-to-one or one-to-many type. We use $t_r = t_p = 0.5$

We use similar formulas for evaluating multiple images as in [13]. For a collection of K images, with K ground truth G^k and K detection D^k. The overall precision and recall scores are:

$$\mathrm{P} = \frac{\sum_k \sum_j Match_D(D_j^k, G^k, t_r, t_p)}{\sum_k |D^k|} \tag{8}$$

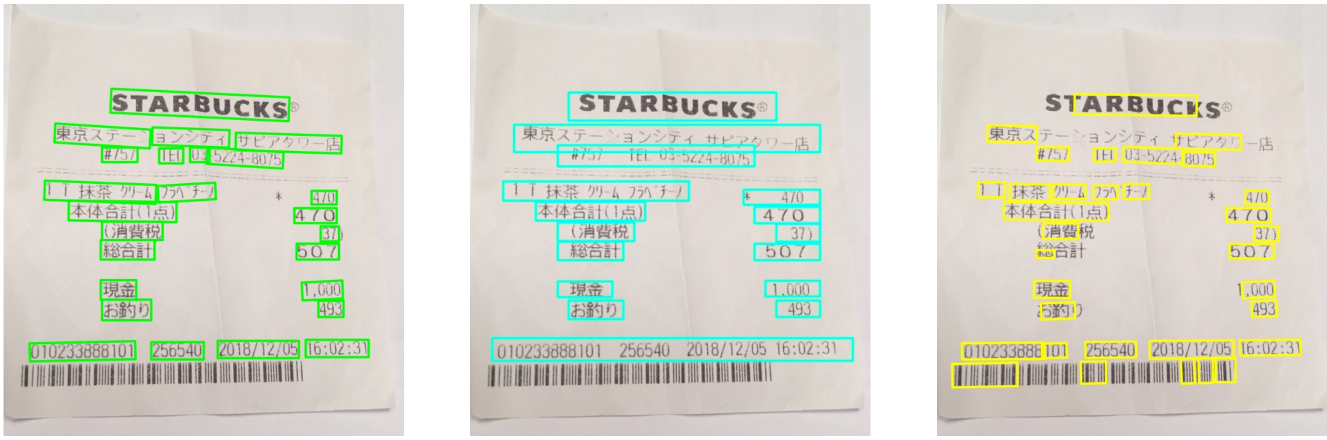
$$\mathrm{R} = \frac{\sum_k \sum_i Match_G(G_i^k, D^k, t_r, t_p)}{\sum_k |G^k|} \tag{9}$$

4.2. Qualitative results and result analysis

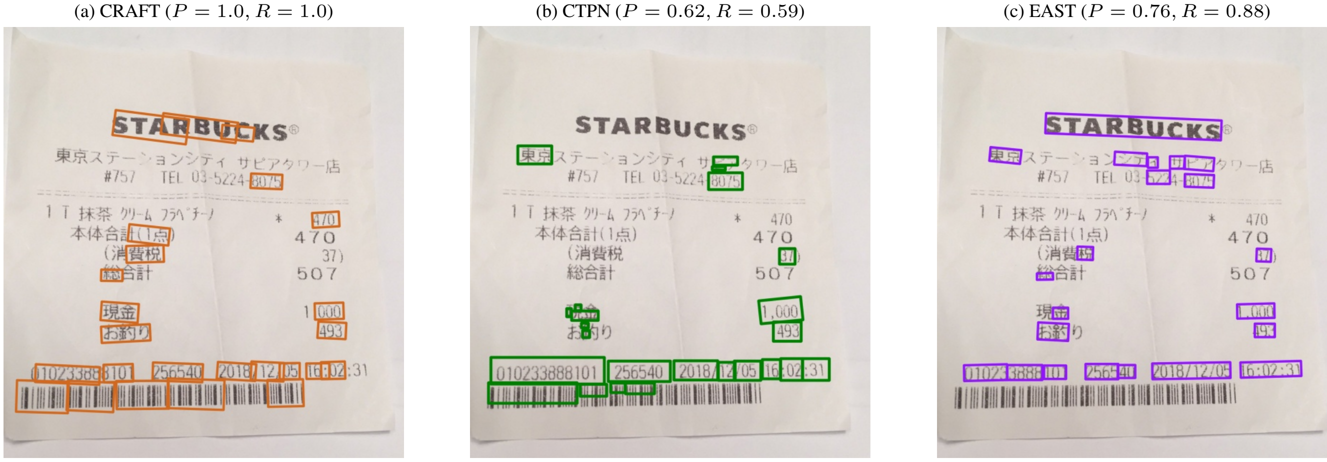
Figure 2 and 3 show the comparison of detected results by six selected state-of-the-art methods for two sample input images from our "BILL" and "CARD" datasets. The ground truth of these images are in Figure 1. The results are cropped in order to hightlight the detected text areas.

Several observations on these results are:

- CRAFT seemingly has the best performance: it detects correctly both horizontal and multi-oriented text areas. Other good performance are from EAST and PSENet.
- CTPN detects horizontal texts quite well but fails for multi-oriented texts.
- PixelLink is said to detect multi-oriented texts but the detected bounding boxes seems either horizontal or having largely redundant spaces; We believe the extra space are caused when combining the eight-orientation link maps in its design.
- CTPN detects longer text areas than other methods; We believe this is due to the connectionist mechanism which tends to connect horizontally close text proposals.
- CTPN, EAST, FOTS, PixelLink can have detected text areas that overlap each other; We believe this overlapping behavior occurs in methods which use post-processing methods like Non-Maximum Suppression (NMS) or contour detection.
- A non-text area such as a portion of logos or barcodes is detected as a text area: CTPN, EAST, FOTS, PixelLink, PSENet; We believe that CRAFT can avoids this false positive because it predicts a segmentation map at character-level instead of pixel-level in other methods.
- CTPN and PixelLink are more likely to detect text areas that merge two or more lines; We believe this is because the neighbor connectivity in the network design (CTPN) or in post-processing step (PixelLink). Other methods make this kind of *mistakes* when two



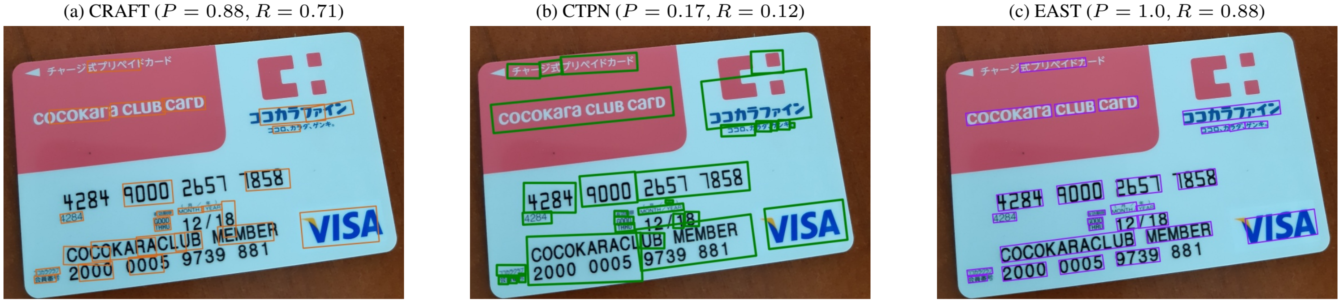
(a) CRAFT ($P = 1.0$, $R = 1.0$) (b) CTPN ($P = 0.62$, $R = 0.59$) (c) EAST ($P = 0.76$, $R = 0.88$)



(d) FOTS ($P = 0.68$, $R = 0.47$) (e) PixelLink ($P = 0.54$, $R = 0.24$) (f) PSENet ($P = 0.55$, $R = 0.35$)

Figure 2: Comparison of the detected text bounding boxes of six SOTA methods for a sample image in our “BILL” dataset

(a) CRAFT ($P = 0.88$, $R = 0.71$) (b) CTPN ($P = 0.17$, $R = 0.12$) (c) EAST ($P = 1.0$, $R = 0.88$)



(d) FOTS ($P = 0.73$, $R = 0.65$) (e) PixelLink ($P = 0.50$, $R = 0.53$) (f) PSENet ($P = 0.89$, $R = 0.82$)

Figure 3: Comparison of the detected text bounding boxes of six SOTA methods for a sample image in our “CARD” dataset

Method	Orientation	Border to text	Broken character	False negative	False positive	Box overlapping	Merged line	Post-processing
CRAFT [1]	multiple	**tight**	**no**	**scarcely**	**scarcely**	**no**	**scarcely**	connected component + minAreaRect
CTPN [9]	horizontal	loose	**no**	sometimes	sometimes	yes	often	NMS + text line formation
EAST [16]	multiple, somewhat horizontal	loose	yes	sometimes	sometimes	yes	**scarcely**	thresholding + NMS
FOTS [7]	multiple	loose	**no**	often	sometimes	yes	**scarcely**	Thresholding + NMS
PixelLink [2]	multiple, somewhat horizontal	loose	yes	often	sometimes	yes	often	contour[1] + minAreaRect
PSENet[12]	multiple	**tight**	yes	sometimes	**scarcely**	**no**	**scarcely**	connected component + scale expansion

[1] Deng *et al*. reported to use connected component in the published paper [2] but we found a contour finding algorithm was actually used in their published source code.

Table 3: Comparison of result quality of six state-of-the-art methods by seven characteristics. The desired qualities are in bold font. There is a noticeable correlation between the post-processing algorithms and the desired qualities. Using "connected component" is likely better than "NMS" or "contour finding".

text lines are too close to each other. This happens with even PSENet, which are designed to avoid merging of adjacent texts using "progressive scale expansion" [12].

- PixelLink can return text areas which are extremely small and are a part of a character of a word; We believe this is because of the behavior of the contour finding in its post-processing step.

We summarize those observations in Table 3.

4.3. Quantitative results: Speed

Table 4 shows a comparison of six state-of-the-art methods in speed of loading and detecting text areas in an input image. The CRAFT method needs a moderate amount of time to load its deep learning model and pretrained weights into the working memory but quickly detects text areas in an image, only 0.23s The EAST method is originally well-known for its speed and still responds quickly (0.29s).

In application development, the loading time can be done once so it does not affect the performance of the application. However, an application is usually expected to respond in less than one second. In this case, the PixelLink and PSENet methods are unlikely applicable. Note that we run all programs on a machine with GPU, which helps speed up computation of deep learning models, for a machine with CPU, we expect the detection time to be much higher.

Method	Library	Loading (s)	Detection (s)
CRAFT [1]	pytorch	10.95	**0.23**
CTPN [9]	keras	0.14	0.88
EAST [16]	tensorflow	1.6	0.29
FOTS [7]	pytorch	2.81	0.45
PixelLink [2]	keras	6.84	6.03
PSENet [12]	pytorch	29.21	1.96

Table 4: Comparison of time to load model and pretrained weights and average time to detect text bounding boxes by SOTA methods. Time unit is in second.

4.4. Quantitative results: Precision and recall

Table 5 and 6 show the precision and recall scores of the selected state-of-the-art methods based on two evaluation protocols, IoU with $threshold = 0.5$ and our modified version W2LEval of the DetEval (see Section 4.1 for more details).

In Table 5, the precision and recall scores of all methods are very low. This is as expected since the text area level of the ground truth in our datasets is line whereas most methods are trained to predict word-level text areas. Using IoU protocol, what we can infer from those results are: (1) CTPN seems to detect longer and larger text areas therefore the overlap with ground truth is larger; (2) other methods will fail badly if the length of the text areas increases such as in CAPTION or NAME datasets.

The W2LEval protocol shows a more fair evaluation

Method	BILL		CAPTION		CARD		NAME		PRODUCT		Average	
	P	R	P	R	P	R	P	R	P	R	P	R
CRAFT [1]	0.25	**0.41**	0.0	0.0	0.28	0.35	0.13	0.27	0.28	**0.53**	0.19	0.31
CTPN [9]	**0.48**	0.4	**0.52**	**0.78**	**0.48**	0.33	**0.31**	**0.3**	**0.38**	0.47	**0.44**	**0.46**
EAST [16]	0.21	0.34	0.0	0.0	0.36	**0.55**	0.09	0.23	0.27	0.49	0.18	0.32
FOTS [7]	0.17	0.26	0.0	0.0	0.24	0.32	0.04	0.10	0.17	0.33	0.12	0.2
PixelLink [2]	0.06	0.15	0.0	0.0	0.18	0.27	0.02	0.05	0.15	0.36	0.08	0.17
PSENet [12]	0.12	0.23	0.0	0.0	0.30	0.46	0.07	0.2	0.16	0.26	0.13	0.23

Table 5: Comparison of precision (P) and recall (R) scores of state-of-the-art methods on our collected datasets using the IoU protocol with $threshold = 0.5$

Method	BILL		CAPTION		CARD		NAME		PRODUCT		Average	
	P	R	P	R	P	R	P	R	P	R	P	R
CRAFT [1]	**0.95**	**0.84**	0.76	**1.0**	0.89	0.77	**0.97**	**0.97**	**0.61**	**0.95**	**0.84**	**0.91**
CTPN [9]	0.45	0.35	0.56	0.81	0.47	0.33	0.33	0.31	0.38	0.46	0.44	0.45
EAST [16]	0.89	0.76	0.80	**1.0**	**0.95**	**0.87**	0.90	0.86	0.56	0.79	0.82	0.86
FOTS [7]	0.85	0.63	**0.81**	0.92	0.74	0.54	0.87	0.73	**0.61**	0.64	0.78	0.69
PixelLink [2]	0.66	0.53	0.77	0.87	0.46	0.44	0.60	0.56	0.57	0.78	0.61	0.64
PSENet [12]	0.79	0.61	**0.81**	0.99	0.91	0.72	0.89	0.78	0.49	0.57	0.78	0.73

Table 6: Comparison of precision (P) and recall (R) scores of state-of-the-art methods on our collected datasets using our W2LEval protocol (a modified version of the DetEval protocol)

of how well each method is as in Table 6. Evaluating by method-wise, all text detection methods are moderately competent in all five demo datasets. Overall, CRAFT yields the highest scores in both precision and recall scores and in most of the datasets. Following CRAFT is EAST, PSENet, FOTS, and PixelLink. CTPN has almost similar scores as being evaluated by IoU in Table 5.

Evaluating by dataset-wise, we can see that the scores vary from one dataset to another. CAPTION is likely the most correctly predicted dataset. Following CAPTION by recall score is NAME, PRODUCT, BILL, and CARD datasets. It seems that the homogeneity of font, font size, direction and the number of incidental texts correlate to those scores. The more homogeneous and simple the fonts are used, the better detection can be done.

Also, CAPTION and PRODUCT datasets have recall scores is higher than precisions by all methods. This can be explained by the number of incidental texts in the data. Because there are no "do not care" label in ground truth data, once those incidental text areas are predicted, they are treated as "false positive" and that causes lower precision.

5. Discussion

Section 4 has shown several observations on the results when applying the state-of-the-art methods in our collected datasets. Overall, we find CRAFT to be the most competent method in both quality of detected areas and quantitative performance. Using only pretrained weights, the CRAFT model can apply to the data which it has never seen and obtains very good precision and recall scores. We believe this achievement is due to its innovative design in training network (character-level instead of pixel-level like other methods). The linking mechanism (neighborhood connectivity) has already been proposed in other forms in many other methods such as CTPN [9] for connecting region proposals, or PixelLink [2] for linking neighbor pixels. However, connecting characters like in CRAFT seems to be the most effective.

We also discussed in Section 4.2 and summarized in Table 3, the post-processing algorithms are strongly related to the performance of the text detection methods. Concretely, defining bounding boxes using connected components is likely to achieve better performance with more desired qualities of the outputs such as tight border or no box overlapping. Moreover, it seems that processing with connected components also provides an economic way to adjust the outputs on any new dataset by modifying the connectivity thresholds. We will investigate this point further in a future work.

A next discussion is regarding the multilingual competence of all methods. The Vietnamese and Japanese texts in our datasets are not seen in the existing datasets on which the state-of-the-art methods are originally trained. The detection quality of most methods show that there are a strong capacity of deep learning models to distinguish textual pixels out of its surrounding background. This seems to be more effective comparing to traditional ways to detect texts using Maximally Stable Extremal Region (MSER) [8] or Stroke Width Transform (SWT) [3]. We are looking forward to seeing new methods that utilize this property while combining more sophisticated post-processing algorithms such like methods before deep learning era.

A final discussion is about how to boost the competence of a SOTA method (e.g., CRAFT) further without retraining the model due to development resources. All SOTA methods use two-step pipeline, consisting of an trainable neural network model and a post-processing step to do final inference of text areas. When retraining a model is not possible, one suggestion is to improve the post-processing algorithms on the outputs of the first step. The post-processing algorithms such as connected components or contour finding or NMS always need fixed thresholding parameters. We can optimize these parameters based on our data or develop an algorithm to find an optimal set of parameters. Besides, we can apply classic computer vision approaches to text detection as post-processing algorithms. Those approaches can be seen in survey works such as [14, 17]. Another suggestion to boost the performance is to ensemble the prediction of the first steps of several methods. A concrete guideline on how to ensemble will need a further study.

6. Conclusion

We have presented several analysis on the performance of six state-of-the-art text detection methods on our five real-world datasets. Our datasets are unconstrainedly collected and manually annotated. The state-of-the-art methods are reimplemented on our machine but using their original pretrained models. The results show that CRAFT is the most competent to most of the datasets than other methods in both speed and quality. We also analyzed which aspect of architecture design affects the performance and found a strong correlation between the post-processing algorithms with the quality of the detected text areas. In future, we would like to expand our experiments on a bigger list of methods and conduct a quantitative study on how much this post-processing contribute to the final performance of a method.

References

[1] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019. 2, 3, 6, 7

[2] D. Deng, H. Liu, X. Li, and D. Cai. Pixellink: Detecting scene text via instance segmentation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2, 3, 6, 7

[3] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2963–2970. IEEE, 2010. 7

[4] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. 1

[5] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE, 2013. 1

[6] J. Liang, I. T. Phillips, and R. M. Haralick. Performance evaluation of document layout analysis algorithms on the uw data set. In *Document Recognition IV*, volume 3027, pages 149–160. International Society for Optics and Photonics, 1997. 4

[7] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5676–5685, 2018. 2, 3, 6, 7

[8] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. 7

[9] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network. In *European conference on computer vision*, pages 56–72. Springer, 2016. 2, 3, 6, 7

[10] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464. IEEE, 2011. 2

[11] K. Wang and S. Belongie. Word spotting in the wild. In *European Conference on Computer Vision*, pages 591–604. Springer, 2010. 1

[12] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9336–9345, 2019. 2, 3, 6, 7

[13] C. Wolf and J.-M. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(4):280–296, 2006. 2, 4

[14] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1480–1500, 2014. 8

[15] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*, 2017. 1

[16] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017. 2, 3, 6, 7

[17] Y. Zhu, C. Yao, and X. Bai. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, 10(1):19–36, 2016. 8