Canny Text Detector: Fast and Robust Scene Text Localization Algorithm

Hojin Cho Myungchul Sung Bongjin Jun
Stradvision, Inc.
{hojin.cho, myungchul.sung, bongjin.jun}@stradvision.com

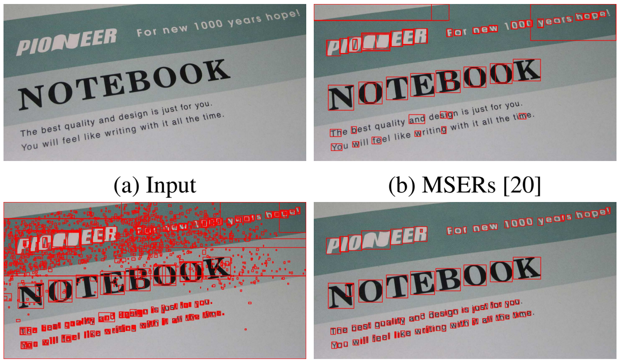
Abstract

This paper presents a novel scene text detection algorithm, Canny Text Detector, which takes advantage of the similarity between image edge and text for effective text localization with improved recall rate. As closely related edge pixels construct the structural information of an object, we observe that cohesive characters compose a meaningful word/sentence sharing similar properties such as spatial location, size, color, and stroke width regardless of language. However, prevalent scene text detection approaches have not fully utilized such similarity, but mostly rely on the characters classified with high confidence, leading to low recall rate. By exploiting the similarity, our approach can quickly and robustly localize a variety of texts. Inspired by the original Canny edge detector, our algorithm makes use of double threshold and hysteresis tracking to detect texts of low confidence. Experimental results on public datasets demonstrate that our algorithm outperforms the state-of-the-art scene text detection methods in terms of detection rate.

1. Introduction

Text in scene images usually conveys valuable information, hence detecting and recognizing scene text has been considered important for a variety of advanced computer vision applications such as image and video retrieval, multilingual translation, and automotive assistance. Especially, as most text recognition applications require texts in images to be localized in advance, there is a significant demand for text detection algorithms that can robustly localize texts from a given scene image.

Previous works for scene text detection have utilized the sliding window method [6, 16, 11, 18] and connected component analysis [8, 5, 35, 37, 38, 12, 21, 22, 23, 28, 41]. The sliding window based methods detect texts of a given scene image by shifting a window onto all locations in multiple scales. This is an exhaustive search, so these methods can achieve high recall rates. However, heavy computations are unavoidable due to the thorough scanning of windows and a large number of candidates can result in a great deal of false positives.


PIONEER
For new 1000 years hope!
NOTEBOOK
The best quality and design is just for you.
You will feel like writing with it all the time.

(a) Input (b) MSERs [20]

(c) ERs after NMS (d) Our localization result

Figure 1. Canny text detector. Compared to the character candidates of MSERs or ERs, our method localizes characters more robustly with less false positives.

On the other hand, connected component based methods first extract character candidates from an input image, and then refine the candidates to suppress non-text candidates. Stroke width transform (SWT) and maximally stable extremal region (MSER) are two representative techniques for connected component analysis, and these methods have achieved outstanding performance in scene text detection. But, common constraints used for refining candidates are considered somewhat restrictive to preserve various true characters, leading to low recall rate in practice.

In this paper, we propose a novel scene text detection algorithm, Canny Text Detector, which takes advantage of the similarity between image edge and text to provide significantly improved detection rate. As edge pixels construct the structural information (i.e., contour) of an object, we observe that cohesive characters compose a word or sentence sharing similar properties such as spatial location, size, color, and stroke width regardless of language. In the original Canny edge detector [4], each edge pixel is first classified as strong edge, weak edge, or non-edge. Then the algorithm employs edge tracking by hysteresis to find connected edges, based on the fact that usually the weak edge pixel coming from true edges are connected to strong edge pixels. Similar to the Canny procedure, we classify *texts* using double threshold and track them by hysteresis to make the best use of plausible text candidates, even if they have low confidence.

Specifically, the proposed Canny text detector is a multi-stage algorithm. We first extract character candidates using a variant of MSER. Then, each candidate is evaluated using an AdaBoost classifier trained with a sort of local binary patterns. The classification step utilizes double threshold to determine strong and weak candidates, and after applying tracking by hysteresis, credible characters are finally selected. The surviving characters are grouped into words or sentences. Experimental results on public datasets demonstrate that our algorithm outperforms the state-of-the-art scene text detection methods in terms of detection rate.

2. Related work

There are a variety of text localization techniques in the literature. The most common approach involves three key components [36]: character candidate extraction, character classification, and text grouping. Grouping text as a set of words or sentences depends on the objective of the algorithm and may involve text line estimation and validation. Existing scene text detection algorithms can be divided into two types based on their character candidate extraction method: (1) sliding window based methods that exhaustively scan windows at all possible locations and scales, and (2) connected component based methods that utilizes character candidates extracted with particular constraints, e.g., consistent stroke width or extremal region.

The sliding window based methods detect text of a given scene image by shifting a window onto all locations in multiple scales [6, 16, 11, 18, 32]. Then for each window, whether the location contains text or not is determined by a classifier that is usually trained with low level features such as image gradients, intensity histogram, and variants of Wavelet coefficients. Although these methods can detect text effectively with high recall rate, their classification can be sensitive to false positives due to the large number of candidates. To suppress false positives, more advanced text/non-text classifiers such as support vector machine and random forest [22, 26, 26] and convolutional neural networks [34, 3, 13, 14] have been also proposed. But due to the heavy computations for intensive window scanning and advanced classification, these approaches are unsatisfactory to real-time applications.

Recent works on scene text detection tend to utilize connected component analysis [8, 5, 35, 37, 38, 12, 21, 22, 23, 28, 41, 40, 39]. In these works, character candidates are first extracted from an input image, where each candidate is a set of pixels sharing similar text properties. The candidates are then refined to suppress non-texts and grouped into final text. Popular techniques for connected component analysis are stroke width transform (SWT) [8, 12] and maximally stable extremal region (MSER) [19, 24], and such methods provide a basis to achieve the notable performance in scene text detection [27, 15].

Recently, Yin et al. [40, 39] proposed several techniques to refine MSERs and improve the robustness for oriented text. Shi et al. [28] utilized geometric information of MSERs for text refinement and grouping. Neumann and Matas [21] used pruning techniques on MSERs to exhaustively search the space of all character sequences. They later included text recognition for end-to-end text reading [22, 23].

Despite the success of connected component analysis methods, we observe that constraints commonly used in previous approaches are not enough to preserve various true characters, leading to low recall rate in practice. Thus, this paper aims to address such limitations.

3. Canny Text Detector

3.1. Criteria for text detection

Given that the prevalent scene text detection procedure is insufficient to achieve high recall rate, we first identify the general criteria that should be considered in text detection, as listed below:

Recall Text detection should localize as many text regions as possible.

Precision The detected results should not contain non-text regions if possible.

Uniqueness Each character detected from the operator should only be marked once.

Compactness The detected region should accurately localize its corresponding character without extra margin.

Similar to the original Canny edge detector [4], we develop a multi-stage algorithm that incorporates the above criteria for effective scene text detection.

3.2. Process overview

Fig. 2 shows the overall process of our text detection algorithm, which is capable of fast and robust localization of scene text. To extract character candidates with better recall rate, we utilize extremal regions (ERs) that are extracted with relatively weak constraints compared to those of the original MSER [20]. Overlapped candidates are reduced to a unique candidate by non-maximum suppression. We then classify the candidates with double threshold as one of strong text, weak text, and non-text. Strong text candidates are included in the final result, and weak text candidates that are connected to the strong texts are only selected

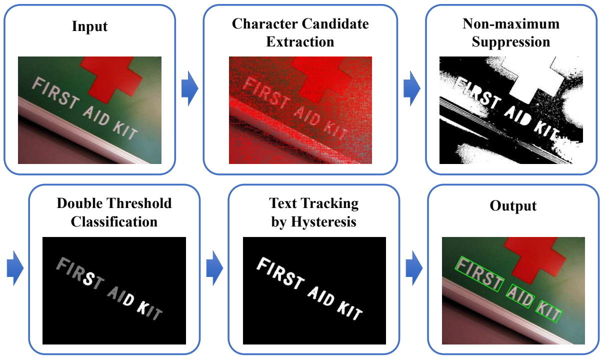

Input
Character Candidate Extraction
Non-maximum Suppression
Double Threshold Classification
Text Tracking by Hysteresis
Output

Figure 2. Overall process of Canny text detector.

by hysteresis. The surviving text candidates are grouped to compose sentence(s).

4. Algorithm Details

In this section, we describe each algorithmic component of Fig. 2 with specific examples as illustrated in Fig. 3.

4.1. Character Candidate Extraction

Many of the previous approaches have adopted MSER [19, 7, 24] to extract character candidates and achieved remarkable performance [27, 15]. However, the constraint for maximum stability is often too strong to embrace various kinds of scene text in practice [40, 31, 30]. So we mitigate the maximum stability constraint and employ only extremal regions (ERs) for better recall to satisfy the *Recall* criterion.

An ER is a set of connected pixels in an image whose intensity values are higher than its outer boundary pixels. Mathematically it is defined as

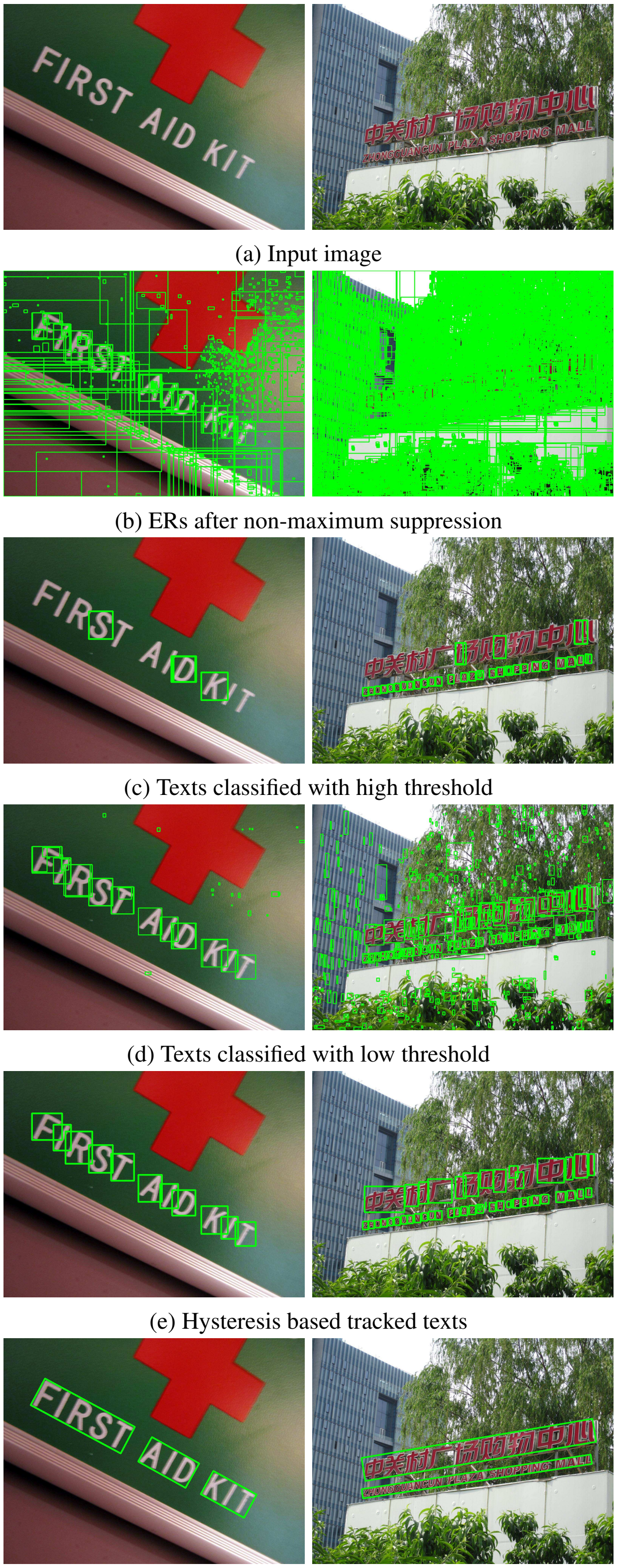
$$R_t = \{x | I(x) > I(y) \quad \forall x \in R_t, \, \forall y \in B(R_t)\}, \tag{1}$$

where x and y are pixel indices of a given single channel image I, t is a threshold value used for extracting the region, and $B(R_t)$ is the set of boundary pixels of R_t. We can easily obtain ERs of an image by thresholding it and building an ER tree using an inclusion relationship between the extracted ERs as described in [7]. The resulting tree is a rooted and directional graph where each node corresponds to one connected component, i.e., extremal region R_t.

Fig. 4 shows an example of an ER tree extracted using the intensity channel of the image shown in Fig. 3a. In this paper, we used six color channels separately to extract ERs, i.e., YCrCb color channels and their inverted channels.

4.2. Non-maximum Suppression

It is well known that MSERs have a large number of repeating components [40]. Since ER is a superset of MSER,



(a) Input image

(b) ERs after non-maximum suppression

(c) Texts classified with high threshold

(d) Texts classified with low threshold

(e) Hysteresis based tracked texts

(f) Results

Figure 3. Intermediate results of the Canny text detection process.

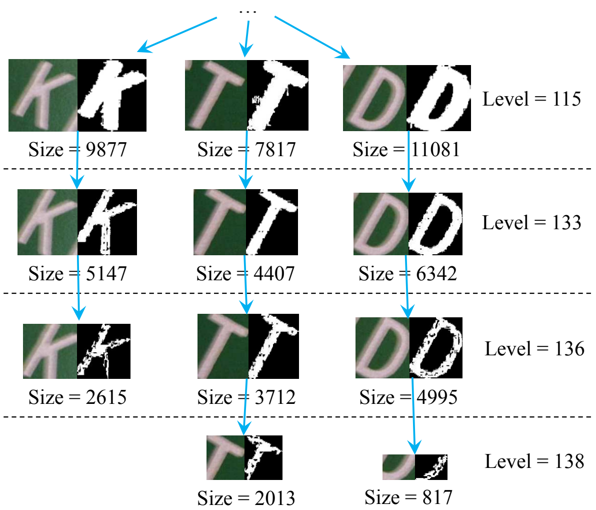


Figure 4. Parts of an ER tree. The input image is shown in the left of Fig. 3a. For each node, the left half shows the cropped image of an ER, and the right half shows its corresponding binary mask of which pixels of the ER are marked as white.

the initial ERs also suffer from the same problem. To guarantee the *Uniqueness* criterion, we suppress the repeating ERs and allow only one ER that has the maximum stability. Note that this process is similar to the subpath partitioning and pruning of Sung et al. [31], but we first find overlapping ERs and then suppress non-maximum ERs with a slightly different stability measure.

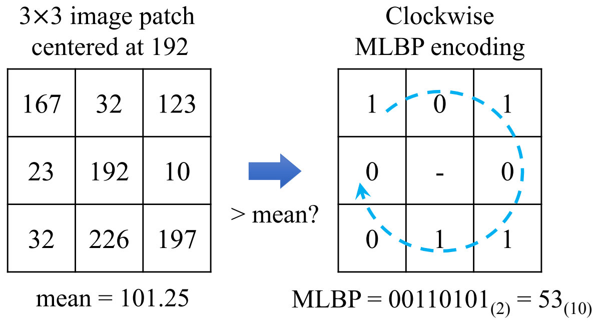
We observe that the repeating component problem mainly occurs because some ERs (i.e., character components) have high contrast and thus are extracted over multiple threshold values (see Fig. 4). To identify the repeating ERs, we use the following measure that estimates overlap between ERs based on the hierarchy of the ER tree:

$$O(R_{t-k}, R_t) = \frac{|R_t|}{|R_{t-k}|}, \tag{2}$$

where R_{t-k} is the parent of R_t in the ER tree, and $|R|$ denotes the bounding box area of R. Note that we do not use R_{t+k} because the ER tree can have multiple children and computing R_{t+k} would be ambiguous. For each node R_t, we count the number of overlaps, n_o, with R_{t-k} for all k such that $O(R_{t-k}, R_t) > 0.7$. Among the overlapping ERs, we remove ERs such that $n_o < 3$ and select the one with the highest stability where the stability is defined as

$$S(R_t) = \frac{(|R_{t-t'}| - |R_t|)}{|R_t|}. \tag{3}$$

We used $t' = 2$ in our implementation. If there exist two or more ERs with the same stability, we choose the one having the smallest area. To further reduce the number of non-texts, we intuitively remove candidates that have too large or too small aspect ratio. After this step, we have character candidates which comply with both *Uniqueness* and *Compactness* criteria. The selected characters through non-maximum suppression are shown in Fig. 3b.



167	32	123
23	192	10
32	226	197

1	0	1
0	-	0
0	1	1

Figure 5. Mean local binary pattern (MLBP) [2].

4.3. Double Threshold Classification

The surviving character candidates are classified into three classes: strong text, weak text, and non-text. For the classification, we train our classifier using AdaBoost [10] and multiple cascades [33] to accelerate the classification speed. The overall structure consists of two blocks of cascaded classifiers, each with a threshold value that satisfies precision of 99.0% and 90.0% in the training set, respectively, corresponding to the high and low threshold values of the original Canny edge detector. Note that the relatively lower precision of 90.0% was intended for finding as many weak texts as possible (i.e., high recall).

Since selection of features has a crucial impact on the classification performance, we use the mean local binary pattern (MLBP) which is known to be robust to illumination and rotation variations [2]. The MLBP is a variant of the local binary pattern [25]. Given a pixel, the average intensity value of 8-connected neighbors in a 3×3 patch is first calculated, and then compared with the intensity value of each pixel excluding the center pixel. If the pixel value is larger then the average value, then the pixel gets '1', otherwise '0'. Then, starting from the left-top pixel and going clockwise, the values are encoded into an 8-bit number, as illustrated in Fig. 5.

For training the English and Chinese classifiers, we gathered about 53,000 and 20,000 positive samples, respectively, together with about 50,000 negative samples (i.e., non-text) for each cascade using a bootstrap process. These samples were normalized to a size of 24×24 in gray-scale.

In our double threshold classification, all candidates goes through the first cascade block, and are classified as strong text or non-strong text. Non-strong text candidates goes through the second cascade block, which in turn classifies them as weak text or non-text. Figs. 3c and d show the classification results with double threshold, i.e., strong texts and weak texts, respectively.

4.4. Text Tracking by Hysteresis

We include the strong text in the final result, as they are classified with high confidence. However, the weak text can be either true text or non-text (e.g., window, leaf, and fence). So they are included if and only if they have similar properties to strong text candidates.

To meet the *Recall* criterion with a high recall rate, we start from each strong text R^s and track its neighborhood text candidates classified as weak text, R^w. Whenever R^w satisfies the similar text properties against R^s, we change the status of R^w to R^s and investigate its neighbors recursively. The properties we used are as follows:

1. The spatial location of R^s and R^w is close enough to be considered as part of the same text. The distance between them is less than twice of the maximum of height and width of R^s.
2. The size (i.e., width and height) of R^s and R^w is similar enough to be considered as part of the same text. In each size dimension, the difference is less than the minimum value between R^s and R^w.
3. The color in the YCrCb color space of R^s and R^w is similar enough to be considered as part of the same text. The difference between them in each channel is less than 25.
4. The ratio between large and small stroke widths of R^s and R^w is less than 1.5.

In our experiments, a variety of text is well tracked in diverse scenes. However, some characters may overlap because of the candidate extraction from different color channels and partial detection (e.g., detecting “l” from “T”) that are not filtered by non-maximum suppression. To address this, we merge overlapping characters after text tracking if their intersection-over-union measure is greater than 0.5. Fig. 3e shows the tracked texts via hysteresis.

4.5. Text Grouping

With double threshold classification and text tracking by hysteresis, we can robustly obtain credible characters. However, some applications require word- or sentence-level localization results in practice. For instance, the robust reading competition (RRC) of the international conference on document analysis and recognition (ICDAR) takes such grouped localization results for evaluation since words can provide more valuable information than individual characters in text reading.

Fortunately, the main advantage of our method is easy grouping. First of all, our method has extracted as many characters as possible, even if they have low confidence. So there is less chance to miss characters in a word, compared to other approaches. Second, as we have tracked character candidates by hysteresis, we can apply almost the same rules for grouping. Specifically, we compare two candidates on spatial location, size, color and aspect ratio using the same threshold values in Sec. 4.4. If they satisfy the properties, then we group them into the same word.

Method	No. of candidates	Recall (%)
All ERs	6,051,331	96.6
MSERs [20]	39,762	53.9
Sung et al. [31]		
Initial ERs	1,729,833	89.6
Refined ERs	93,357	87.7
Our method		
ERs after NMS	629,932	95.1
Final characters	8,121	87.4

Table 1. Evaluation of character-level recall on the ICDAR 2011 test set.

To provide compact bounding boxes as output, we compute the minimum-area encasing rectangle [9]. Unlike the previous approaches [35, 17, 39], we do not estimate the bottom or center line of characters. Instead, we estimate the smallest rectangle that encloses grouped characters in the 2D image space using the 2D coordinates of character pixels. The final grouping results of the proposed method are shown in Fig. 3f.

5. Experimental Results

We implemented our method using C/C++. Our testing environment is a PC running MS Windows 7 64bit version with Intel Core i7 CPU of 4.00GHz. In this section, we quantitatively evaluate the proposed algorithm in terms of character-level recall rate and text-level localization performance on the most widely used public datasets: ICDAR 2011 RRC [27], ICDAR 2013 RRC [15], and a multilingual dataset [26] that contains both English and Chinese. Particularly, we use the images of “Challenge 2: Reading Text in Scene Images” in the ICDAR RRC.

Table 1 shows a quantitative comparison of character-level recall on the ICDAR 2011 dataset [27] with the state-of-the-art candidate extraction method proposed by Sung et al. [31]. We obtained the ground truth data from the author that contains manually specified character-level bounding boxes for each image. The total number of images and characters in the test set are 255 and 6,309, respectively. Given ground truth bounding boxes, we determine the localized result as a correct detection if the intersection-over-union measure between a detected region and the ground truth region is over 0.5. Our method quickly reduces the number of candidates using non-maximum suppression in the ER tree

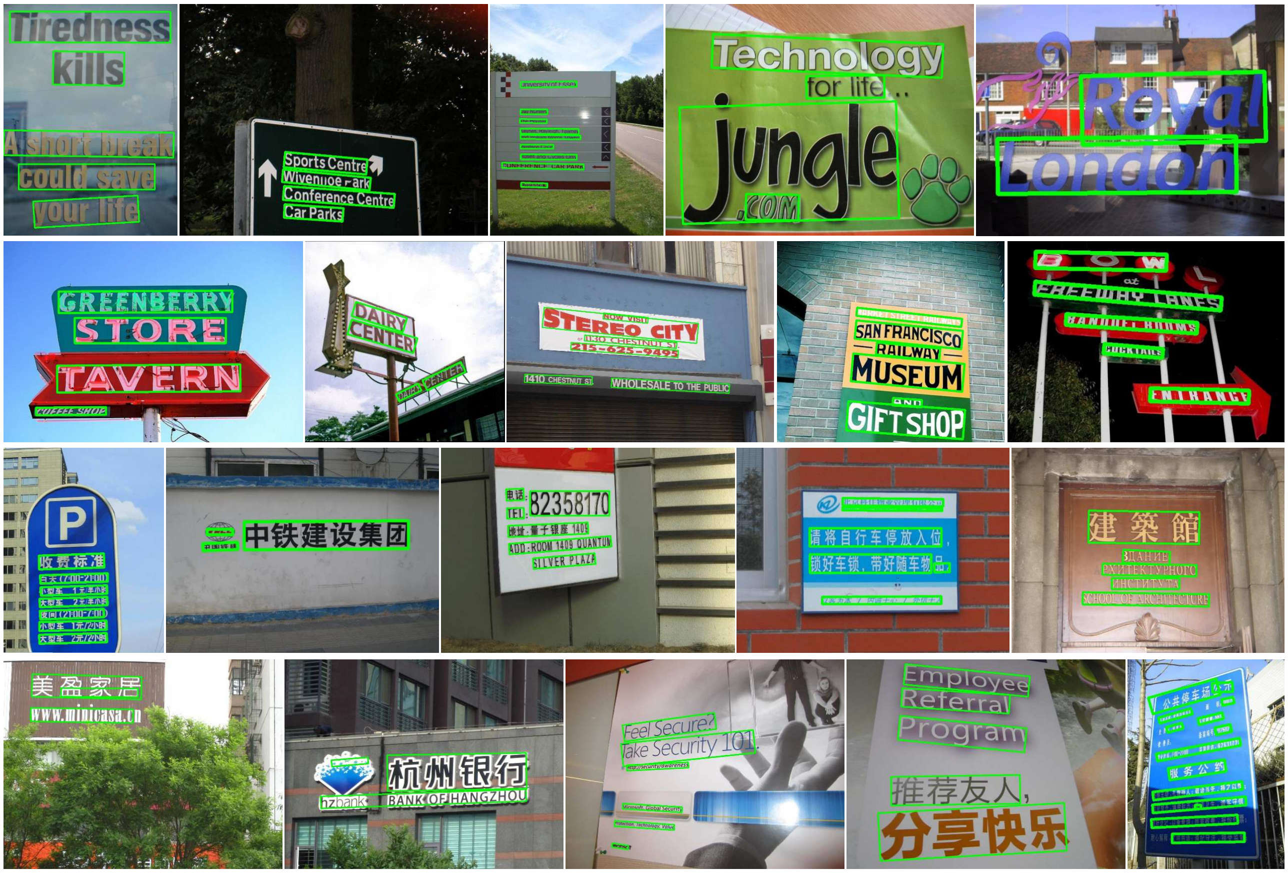


Figure 6. Sample results on scene text detection. We take the input images from publicly available datasets: the ICDAR 2013 RRC, the multilingual dataset proposed by [26], MSRA-TD500, and HUST-TR400. Our results are marked in green bounding boxes.

and results in almost one third of the initial ERs compared to [31]. It is worth mentioning that our final localization results have reduced more than 90% of irrelevant candidates while preserving a comparable recall rate to the refined ERs of Sung et al. [31] that still require further processing such as classification.

We also estimated the running time of our method with the ICDAR 2011 test set. The average image size of the dataset is about 1,145 by 886 pixels. On average, our method took 0.13 seconds to process one image (i.e., character candidate extraction, non-maximum suppression, double threshold classification, and text tracking by hysteresis).

We also evaluated our method on the ICDAR 2013 dataset [15]. Table 2 shows the quantitative results provided by the online competition website. The winning algorithm of the ICDAR 2013 RRC (Challenge 2), proposed by Yin et al. [40], achieved a harmonic mean of 75.89% while our approach obtains 82.17%. The increased recall of ours is mainly due to the double threshold classification and text tracking by hysteresis.

Method	Recall	Precision	Hmean
Shi et al. [29]	62.85	84.70	72.16
Bai et al. [1]	68.24	78.89	73.18
Yin et al. [39]	65.11	83.98	73.35
Neumann and Matas [22]	64.84	87.51	74.49
Yin et al. [40]	66.45	88.47	75.89
Zamberletti et al. [41]	70.–	86.–	77.–
Tian et al. [32]	75.89	85.15	80.25
Sung et al. [31]	74.23	**88.65**	80.80
Our method	**78.45**	86.26	**82.17**

Table 2. Evaluation on the ICDAR 2013 competition on robust reading test set.

To validate our method on another language, we use the multilingual dataset proposed by Pan et al. [26]. The training set contains 248 images and the testing set contains 239 images. Given the ground-truth text region set GT and the detected text region set DT, the precision rate p of each detected region rectangle dt and the recall rate r of each

Method	Recall	Precision	Hmean
Pan et al. [26]	65.9	64.5	65.5
Baseline	67.2	78.6	72.4
Yin et al. [40]	68.5	82.6	74.6
Tian et al. [32]	78.4	84.7	81.4
Our method	**93.5**	**93.1**	**93.3**

Table 3. Evaluation on the multilingual test set.

ground-truth text region rectangle gt are defined as

$$p(dt) = \max_{gt \in GT}[m(dt, gt)], \tag{4}$$

$$r(gt) = \max_{dt \in DT}[m(dt, gt)], \tag{5}$$

where $m(dt, gt)$ is the intersection-over-union measure between dt and gt [26]. The final precision and recall rates are the average of p and r for all dt and gt, respectively. As shown in Table 3, the proposed algorithm has dramatically improved recall, precision, and their harmonic mean. The average running time of our method was 0.08 seconds on our PC. Although previous works have provided timing results for this dataset, computing environments are all different so fair comparison is not possible. We provide our running time for future reference.

Fig. 6 shows several sample results taken from MSRA-TD500 and HUST-TR400 public datasets as well as the datasets used in this paper. Regardless of the language, the Canny text detector works robustly for localizing a wide range of texts, even if there exist noise, blur, and disturbing textures such as windows, tree leaves, and so on. One of the merits of our algorithm is the fast speed and this is demonstrated on our demo website[1]. More text detection examples are provided in the supplementary material.

6. Discussion and Future Work

Our Canny text detector delivers a fast and robust algorithm for scene text detection. The proposed approach is intuitive and easy to implement since we do not involve complex operations such as image optimization. Instead, the overall process is similar to the famous Canny operator that has been proven to be effective in the edge detection literature. Despite the simplicity of our algorithm, experiments on widely used datasets demonstrate that the proposed method can effectively localize texts in practice. The key to Canny text detector is double threshold classification and text tracking by hysteresis. We expect such effective detection framework can be adopted for other detection applications. In the following, we discuss some issues related to our approach.

Effect of dataset When compared to the state-of-the-art text detection methods, our method performs well in terms of recall score. This is not because we simply used more training images than others. To clarify this, we examined existing approaches [22, 40, 31] with the same classifier we trained with our training images. However, whatever candidate extraction method is used, using character candidates classified with only a single threshold performed poorly (i.e., low recall if a high threshold value is used, and low precision if a low one is used). We do believe that our new framework can improve the detection rate as well as interoperate with existing methods. For follow-up researchers, we will provide our training dataset upon request by email.

Interoperability with existing localization methods As prevalent text detection approaches already use character classification, the essence of the proposed method (i.e., double threshold classification and text tracking) can be incorporated for interoperability to other methods without much difficulty.

Future work The fast speed and accurate localization of the Canny text detector lowers the barrier to develop a real-time end-to-end text reading system. Although there exist a bunch of text recognition algorithms available in practice [36], recent techniques employs a large amount of convolution operations for accuracy. Thus, we first need to optimize the running speed or develop an efficient recognition algorithm. In future, we will also explore along this direction to develop video algorithms.

References

[1] B. Bai, F. Yin, and C. L. Liu. Scene text localization using gradient local correlation. In *Proc. ICDAR 2013*, pages 1380–1384, 2013.

[2] G. Bai, Y. Zhu, and Z. Ding. A hierarchical face recognition method based on local binary pattern. In *Proc. Congress on Image and Signal Processing (CISP)*, pages 610–614, 2008.

[3] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. PhotoOCR: Reading text in uncontrolled conditions. In *Proc. ICCV 2013*, pages 785–792, 2013.

[4] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis Machine Intelligence*, 8(6):679–698, June 1986.

[5] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Proc. ICIP 2011*, pages 2609–2612, 2011.

[6] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *Proc. CVPR 2004*, pages 366–373, 2004.

[7] M. Donoser and H. Bischof. Efficient maximally stable extremal region (MSER) tracking. In *Proc. CVPR 2006*, pages 553–560, 2006.

[1] http://stradvision.com/demo.html

[8] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Proc. CVPR 2010*, pages 2963–2970, 2010.

[9] H. Freeman and R. Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Commun. ACM*, 18(7):409–413, July 1975.

[10] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[11] S. M. Hanif and L. Prevost. Text detection and localization in complex scene images using constrained adaboost algorithm. In *Proc. ICDAR 2009*, pages 1–5, 2009.

[12] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proc. CVPR 2013*, pages 1241–1248, 2013.

[13] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced MSER trees. In *Proc. ECCV 2014*, pages 497–511, 2014.

[14] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision (IJCV)*, pages 1–20, 2015.

[15] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. de las Heras, et al. ICDAR 2013 robust reading competition. In *Proc. ICDAR 2013*, pages 1484–1493, 2013.

[16] K. I. Kim, K. Jung, and J. H. Kim. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Trans. Pattern Analysis Machine Intelligence*, 25(12):1631–1639, Dec 2003.

[17] H. I. Koo and D. H. Kim. Scene text detection via connected component clustering and nontext filtering. *IEEE Trans. Image Processing*, 22(6):2296–2305, June 2013.

[18] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. L. Yuille, and C. Koch. Adaboost for text detection in natural scene. In *Proc. ICDAR 2011*, pages 429–434, 2011.

[19] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC 2002*, pages 384–396, 2002.

[20] 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.

[21] L. Neumann and J. Matas. Text localization in real-world images using efficiently pruned exhaustive search. In *Proc. ICDAR 2011*, pages 687–691, 2011.

[22] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Proc. CVPR 2012*, pages 3538–3545, 2012.

[23] L. Neumann and J. Matas. On combining multiple segmentations in scene text recognition. In *Proc. ICDAR 2013*, pages 523–527, 2013.

[24] D. Nister and H. Stewenius. Linear time maximally stable extremal regions. In *Proc. ECCV 2008*, pages 183–196, 2008.

[25] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis Machine Intelligence*, 24(7):971–987, Jul 2002.

[26] Y.-F. Pan, X. Hou, and C.-L. Liu. A hybrid approach to detect and localize texts in natural scene images. *IEEE Trans. Image Processing*, 20(3):800–813, 2011.

[27] A. Shahab, F. Shafait, and A. Dengel. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In *Proc. ICDAR 2011*, pages 1491–1496, 2011.

[28] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao. Scene text detection using graph model built upon maximally stable extremal regions. *Pattern Recognition Letters*, 34(2):107–116, 2013.

[29] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang. Scene text recognition using part-based tree-structured character detection. In *Proc. CVPR 2013*, pages 2961–2968, 2013.

[30] L. Sun, Q. Huo, W. Jia, and K. Chen. A robust approach for text detection from natural scene images. *Pattern Recognition*, 48(9):2906–2920, 2015.

[31] M.-C. Sung, B. Jun, H. Cho, and D. Kim. Scene text detection with robust character candidate extraction method. In *Proc. ICDAR 2015*, pages 426–430, 2015.

[32] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan. Text flow: A unified text detection system in natural scene images. In *Proc. ICCV 2015 (to appear)*, 2015.

[33] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, 2004.

[34] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *Proc. ICCV 2011*, pages 1457–1464, 2011.

[35] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Proc. CVPR 2012*, pages 1083–1090, 2012.

[36] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. *IEEE Trans. Pattern Analysis Machine Intelligence*, 37(7):1480–1500, July 2015.

[37] C. Yi and Y. Tian. Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification. *IEEE Trans. Image Processing*, 21(9):4256–4268, 2012.

[38] C. Yi and Y. Tian. Text extraction from scene images by character appearance and structure modeling. *Computer Vision and Image Understanding*, 117(2):182–194, 2013.

[39] X. Yin, W. Pei, J. Zhang, and H. Hao. Multi-orientation scene text detection with adaptive clustering. *IEEE Trans. Pattern Analysis Machine Intelligence*, 37(9):1930–1937, Sept 2015.

[40] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao. Robust text detection in natural scene images. *IEEE Trans. Pattern Analysis Machine Intelligence*, 36(5):970–983, May 2014.

[41] A. Zamberletti, L. Noce, and I. Gallo. Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions. In *Proc. Int'l Workshop on Robust Reading (in ACCV 2014)*, pages 91–105, 2014.