SCENE TEXT DETECTION BASED ON COMPONENT-LEVEL FUSION AND REGION-LEVEL VERIFICATION

Guanghan Ning, Tony X. Han, and Zhihai He

Department of Electrical and Computer Engineering
University of Missouri, Columbia, MO 65211

ABSTRACT

In this paper, we present a novel scene text detection method that combines the advantages of component-based methods and region-based methods, while overcoming their inherent limitations. We first extract text regions as candidates, and then aggregate these text components in these regions into words and text lines. To separate non-text components in the background from text components, we perform both character-level filtering and word-level classification with a trained linear SVM (support vector machine) classifier. Our extensive experiments on ICDAR2003 and ICDAR2011 datasets have shown that our method outperforms the state-of-the-art methods in text detection.

Index Terms— Scene text detection, adaptive local binarization, maximally stable extremal region (MSER), patch classification.

1. INTRODUCTION

Detecting and recognizing text from natural scene images remains an open and challenging problem in computer vision with many interesting applications in intelligent video analysis, scene understanding, human-computer-environment interactions, etc. Compared to traditional document OCR, text recognition from natural images is much more challenging because it needs to deal with texts with large variations in patterns, fonts, colors, scales, and orientations. It also needs to handle complex and cluttered background, as well as variations and distortion caused by illumination.

Currently, existing methods for scene text detection can be categorized into two classes: texture (or region)-based and component-based methods. Texture-based methods assume that text regions usually have unique texture characteristics from other regions, while component-based methods assume that text components tend to have similar properties, such as font, color, and stroke width.

Region-based approaches focus on efficient binary classification of small image patches. They perform local decisions on a sliding window. Specifically, a feature vector extracted from each local region is fed into a classifier for estimating the likelihood of text. It has been observed that these types of small image patches with unknown knowledge of text properties such as scales, colors, and orientations are quite challenging for accurate and robust classification. In component-based approaches, connected components are extracted first, and then non-text components are pruned based on heuristic rules or with trained classifiers. The majority of background pixels are expected to be discarded using low-level features or filters, and the remaining pixels can be used to construct component candidates based on properties of text, such as consistency of stroke width [1, 2], color uniformity [2], and size similarity [1]. One major problem with these types of component-based methods is that the construction of component is sensitive to image noise and distortion. Furthermore, the subsequent filters or classifiers may not be effective enough for removing non-text components.

In this work, we study how these two approaches can be coupled together to achieve more efficient text detection. Our proposed method aims to combine the advantages of both texture-based and component-based methods, while overcoming their inherent limitations. We first develop component-based methods to perform over-extraction of text-like regions as candidates, making sure that all text regions are included in these candidates. We then develop a region-based method to filter out the vast amount of non-text components. Without the need to use a sliding window or multi-scale image scanning, the proposed method is much less computationally expensive when compared to existing region-based methods [3].

The remainder of the paper is organized as follows. Section 2 presents the overall system for text detection. Section 3 to 7 illustrates the system in more details. Section 8 shows that our method is very competitive being among the state-of-the art algorithms in the ICDAR dataset. Finally, Section 9 concludes the paper.

2. ALGORITHM OVERVIEW

Fig. 1 provides an overview of the proposed method. The algorithm consists of four major steps: (1) text component extraction, (2) text component aggregation, (3) character-level filtering, and (4) word-level filtering. The first two steps, text region extraction and text component aggregation, aim at detecting text word candidates. The remaining two steps, character-level and word-level filtering, aim to suppress and remove non-text candidates. Specifically, an input image first goes through two different local connected component detectors, where two text region masks are generated and fused together to achieve efficient text component extraction. These text components are candidate regions for text. Geometric filtering coupled with stroke width uniformity test is used to remove non-text components. We observe that some non-text components will be connected with text. To address this issue, we fuse an edge map to detach the components. We then develop a component aggregation scheme to merge text components (characters) into words and text lines, and then segment words from each other based on their spatial relationship. We extract features from each word-level candidate (an image patch) and train a model to classify if the candidate is text or from the background.

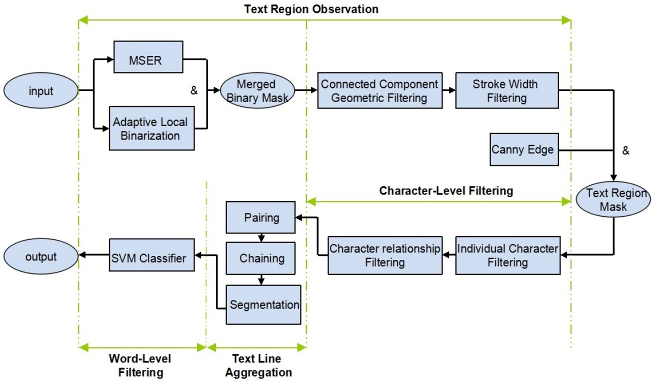


Fig. 1: Overview of the proposed algorithm for text detection from natural scene images

3. TEXT COMPONENT EXTRACTION

In this work, we use a combination of MSER (maximally stable extremal regions) and adaptive local binarization methods to extract text components. The task of MSER is to determine the candidate text image regions. During our experiments, we purposefully lower down the threshold during MSER analysis to maximize the detection probability of text regions. Certainly, this will generate a large number of non-text background regions. It has been observed that MSER is sensitive to image blur. To address this issue, we use local adaptive binarization [4] and connect component analysis as a complementary approach for text component extraction. Traditional binarization methods based on global thresholding are unable to achieve efficient binarization for different image content. Local adaptive binarization methods [4, 5, 6], on the other hand, examine local statistics in the neighborhood and determine the binarization threshold at the pixel level. It works well on both small and large text region extraction, and it is robust to illumination change.

Each of these two approaches, the MSER and adaptive local binarization, generates a binary mask indicating if an image region belongs the text or not. We use the *and* operator to merge these two binary masks. We then apply connected component analysis to extract text components. At this stage, each component corresponds to a character. Basic properties of characters, such as area, eccentricity, and solidity, are used to filter out non-character components [7]. Specifically, components of extremely small areas are considered noise. The eccentricity condition indicates that a character is not supposed to be over flat, nor too thin. The minimum solidity implies that text characters tend to fill the space.

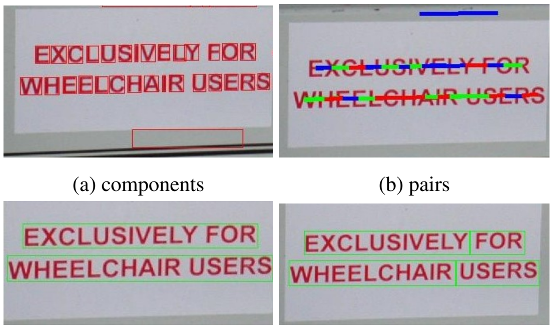
To further remove non-character components, we also use the stroke width properties of text. Recent studies [1, 2, 8] demonstrate that stroke width information is important and effective for text detection. We apply the stroke width transform to the binary mask image obtained with MSER and adaptive local binarization. Each pixel is assigned a value which represents it local stroke width, which is the minimum width of the text region at the pixel location. One important property of stroke width is that pixels within a character have similar stroke width. Let $w(x, y)$ be the stroke width at pixel (x, y) inside a text region R. Let $m(R)$ be average of $w(x, y)$, $(x, y) \in R$, and $\sigma(R)$ be the standard deviation of $w(x, y)$. We use the following criteria

$$\frac{\sigma(R)}{m(R)} \geq \delta, \tag{1}$$

to filter out non-character regions. Here, δ is a control parameter. It should be noted that this filtering method is invariant to scale changes. Because some text components are connected with non-text components, they need to be separated first in order to further apply stroke width to filter out non-text components. Otherwise, text components may be mistakenly eliminated if they are connected to non-text components.

4. WORD DETECTION

The task of word detection is to group characters into words, or text lines. Morphological operators such as opening and closing have been used to aggregate text components into words [7], but they have the drawback of over-aggregation of small character regions. Since the text sizes are not uniform across all the test images, it is difficult to choose the right kernel size for morphological operations. In this work, we develop a text aggregation method which exploits the local properties between two text components. Text components with similar properties will be merged. Although the text size may change significantly from one image to another, or from one image region to another, the sizes and stroke width of two adjacent text components remain the same. We merge two neighboring characters into one pair if they share similar sizes and stroke width. Overlapping pairs are linked into text lines. Text-like noises may survive the individual property check, but most of them do not aggregate into lines as real texts. In this way, the pairing and chaining procedure can efficiently filter out some non-text components. Since some words may be within the same text line, we developed a segmentation method that examines the spatial statistics of the pairs, in order to segment word candidates when necessary. Fig. 2 shows one example of how text components are paired and linked together into text lines.



(a) components (b) pairs

(c) before segmentation (d) after segmentation

Fig. 2: The text line aggregation process

5. WORD-LEVEL FILTERING: TEXT AND NON-TEXT CLASSIFICATION

In the previous steps, individual components are grouped and then linked into word component. During our experiments, we observe that a large portion of these word components are not text and come from the background. To address this issue, we propose to train a classifier to determine if a word component (an image region) is text or not. It should be noted that this type of classification can operate efficiently at the word level since a word typically corresponds to a large image region with sufficient statistics. However, the classification scheme may not work efficiently at the previous stages of character-level. Unlike the previous region-based text detection schemes, the proposed word-level classification operates on the whole word image patch, without the need to scan over images, which often involves very high computational complexity.

To characterize the word image regions, we use HOG (histogram of oriented gradients) and LBP (local binary pattern) features. It has been observed that the HOG feature is very efficient in capturing the edge and shape information of text and differentiating them from the background [9]. The LBP feature is efficient in capturing the difference of texture characteristics between the text and background. We compute the HOG and LBP for each pixel and scan the image patch with a small cell. Each cell produces a HOG-LBP feature vector. Following the bag-of-words (BOW) model, we use k-mean clustering to build a codebook of 1500 codewords. We then compute the histogram of these codewords for all cells in the image patch. The normalized histogram vector of size 1500 is used as the feature vector for classification with linear SVM (support vector machine). We use the approach outlined in the previous sections to generate a large number of word candidate patches. We then manually separate them into text and non-text classes for training. In total, we have created a training set with 3000 positive and 7000 negative samples. Using 70% of them as training and 30% of them as testing data with random partition, we have achieved an average accuracy of $> 97\%$ in text and non-text classification.

6. EXPERIMENTAL ANALYSIS

In this work, we are using the dataset from ICDAR 2003 and ICDAR 2011 to evaluate our performance on the scene text localization problem. In order to evaluate the effects of parameters, we use the standard performance metrics, including precision, recall, and f measure. Cluster centers N and MSER threshold can be utilized in controlling the overall precision-recall. The cluster number N is set to 1500 when comparing with other methods. Both precision and recall rate will drop when N gets smaller. In general, precision and recall are used to measure a retrieval system as follows. For a given query, we have a ground-truth set of targets T and the set returned by the system under test, which we call estimates, E. We define the match m_p between two rectangles as the area of intersection divided by the area of the minimum bounding box containing both rectangles. The best match $m(r, R)$ for a rectangle r in a set of rectangles R is defined as:

$$m(r, R) = \max m_p(r, r^{'}) \| r^{'} \in R \tag{2}$$

Our definitions of precision and recall are:

$$p = \frac{\sum r_{e \in E} m(r_e, E)}{|T|} \tag{3}$$

$$r = \frac{\sum r_{t \in T} m(r_t, E)}{|T|} \tag{4}$$

We adopt the standard f measure and we set α to 0.5 to provide equal weights to precision and recall:

$$f = \frac{1}{\alpha/p + (1-\alpha)/r} \tag{5}$$

We have plotted the precision-recall curve for the ICDAR2011 dataset in Fig. 3. From Tables 1 and 2, we can see that our method outperforms other algorithms.

7. CONCLUSION

In this paper, we have developed a novel scene text detection algorithm that couples component-based with region-based

Table 1: Performance comparison of text detection algorithms on ICDAR 2003 test dataset. Our system 1 includes segmentation of text word while system 2 does not include this step.

Algorithm	Precision	Recall	f-measure
Our system 1	**0.80**	**0.68**	**0.74**
Our system 2	0.76	0.62	0.68
Yao et al. [10]	0.69	0.66	0.67
Epshtein et al. [1]	0.73	0.60	0.66
Yi et al.[11]	0.71	0.62	0.62
Becker et al.[12]	0.62	0.67	0.62
Chen et al.[13]	0.60	0.60	0.58
Ashida [14]	0.55	0.46	0.50

Table 2: Performance comparison of text detection algorithms on ICDAR 2011 test dataset.

Algorithm	Year	Precision	Recall	F
Our system	-	**0.83**	0.67	**0.74**
Huang et al. [2]	2013	0.82	**0.75**	0.73
Neumann et al. [8]	2013	0.79	0.66	0.72
Neumann et al. [15]	2012	0.73	0.65	0.69
Yi and Tian [16]	2013	0.76	0.68	0.67
Gonzalez et al. [17]	2012	0.73	0.56	0.63
Yi and Tian [11]	2011	0.67	0.58	0.62
Neumann et al. [18]	2011	0.69	0.53	0.60

methods. We have not only used geometric features and text-specific features like stroke width, but also used machine learning techniques such as SVM and a bag-of-words model with HOG and LBP features. Both character-level and word-level filtering are exploited. Spatial information is considered by aggregating adjacent text components. We have conducted experiments on ICDAR 2003 and 2011 datasets which showed that our method yields the state-of-the-art performance.

The major contribution of this paper lies in the following two major aspects. First, our algorithm combines the advantages of component-based and region-based methods for text detection. Second, we couple text component detection and word-level patch classification to achieve highly efficient text detection.

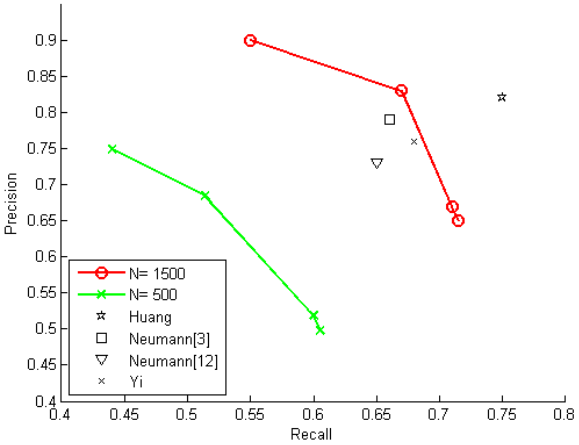


Fig. 3: The precision-recall curves are derived with different MSER parameters. The number of text candidates drops when MSER is stricter, resulting in the decrease of recall rate, but holds comparatively higher precision rate.

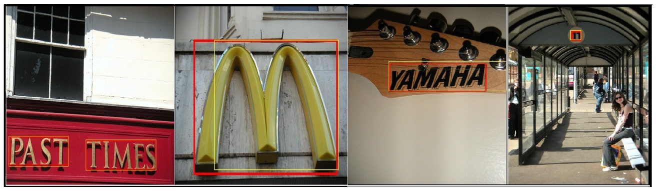


Fig. 4: Detection examples from the ICDAR2003 text dataset. Bounding rectangles of the ground truth are indicated in red while the detection results in yellow.

8. REFERENCES

[1] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2963–2970, 2010.

[2] Weilin Huang, Zhe Lin, Jianchao Yang, and Jue Wang, "Text Localization in Natural Images Using Stroke Feature Transform and Text Covariance Descriptors," *2013 IEEE International Conference on Computer Vision*, pp. 1241–1248, Dec. 2013.

[3] Kwang In Kim, Keechul Jung, and Jin Hyung Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 12, pp. 1631–1639, Dec 2003.

[4] Xiaoqian Liu, Ke Lu, and Weiqiang Wang, "Effectively localize Text in Natural Scene Images," , no. Icpr, pp. 1197–1200, 2012.

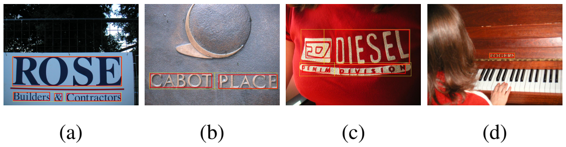


Fig. 5: Detection examples from the ICDAR2011 text dataset.

[5] Bilal Bataineh, Siti Norul Huda Sheikh Abdullah, and Khairuddin Omar, "An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows," *Pattern Recognition Letters*, vol. 32, no. 14, pp. 1805 – 1813, 2011.

[6] T. Romen Singh, Sudipta Roy, O. Imocha Singh, Tejmani Sinam, and Kh. Manglem Singh, "A new local adaptive thresholding technique in binarization," *CoRR*, vol. abs/1201.5227, 2012.

[7] Huizhong Chen, Sam S. Tsai, Georg Schroth, David M. Chen, Radek Grzeszczuk, and Bernd Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions.," in *ICIP*, Benot Macq and Peter Schelkens, Eds. 2011, pp. 2609–2612, IEEE.

[8] Lukas Neumann and Jiri Matas, "Scene text localization and recognition with oriented stroke detection," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[9] Chucai Yi, Xiaodong Yang, and YingLi Tian, "Feature representations for scene text character recognition: A comparative study," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, Aug 2013, pp. 907–911.

[10] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu, "Detecting texts of arbitrary orientations in natural images," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 8, pp. 1083–1090, June 2012.

[11] Chucai Yi and YingLi Tian, "Text string detection from natural scenes by structure-based partition and grouping," *Image Processing, IEEE Transactions on*, vol. 20, no. 9, pp. 2594–2605, Sept 2011.

[12] S.M. Lucas, "Icdar 2005 text locating competition results," in *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, Aug 2005, pp. 80–84 Vol. 1.

[13] Xiangrong Chen and A.L. Yuille, "Detecting and reading text in natural scenes," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, June 2004, vol. 2, pp. II–366–II–373 Vol.2.

[14] S M Lucas, A Panaretos, L Sosa, A Tang, S Wong, and R Young, "ICDAR 2003 Robust Reading Competitions," , no. Icdar, 2003.

[15] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 3538–3545.

[16] Chucai Yi and Yingli Tian, "Text extraction from scene images by character appearance and structure modeling," *Computer Vision and Image Understanding*, vol. 117, no. 2, pp. 182 – 194, 2013.

[17] A. Gonzalez, L.M. Bergasa, J.J. Yebes, and S. Bronte, "Text location in complex images," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, Nov 2012, pp. 617–620.

[18] L. Neumann and J. Matas, "Text localization in real-world images using efficiently pruned exhaustive search," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, Sept 2011, pp. 687–691.