ICIP 2015
IEEE International Conference on Image Processing
27-30 SEPTEMBER 2015, QUÉBEC CITY, CANADA

Mizzou
University of Missouri - Columbia

Scene Text Detection Based on Component-Level Fusion and Region-Level Verification

Authors:

Guanghan Ning,

Tony(Xu) Han,

Zhihai(Henry) He

The Problem

What is Scene Text Detection?

Scene Text Detection is the process of localizing texts in natural scene images, in contrast to texts in scanned documents.

The **significance** lies in two aspects:

1. It is an important prerequisite for many content-based image analysis tasks, as it provides more descriptive and abstract information beyond intuitive perception of other objects.

2. Other potential applications include assistive navigation, scene understanding, etc.

Dataset & Evaluation

International Conference on Document Analysis and Recognition(ICDAR)

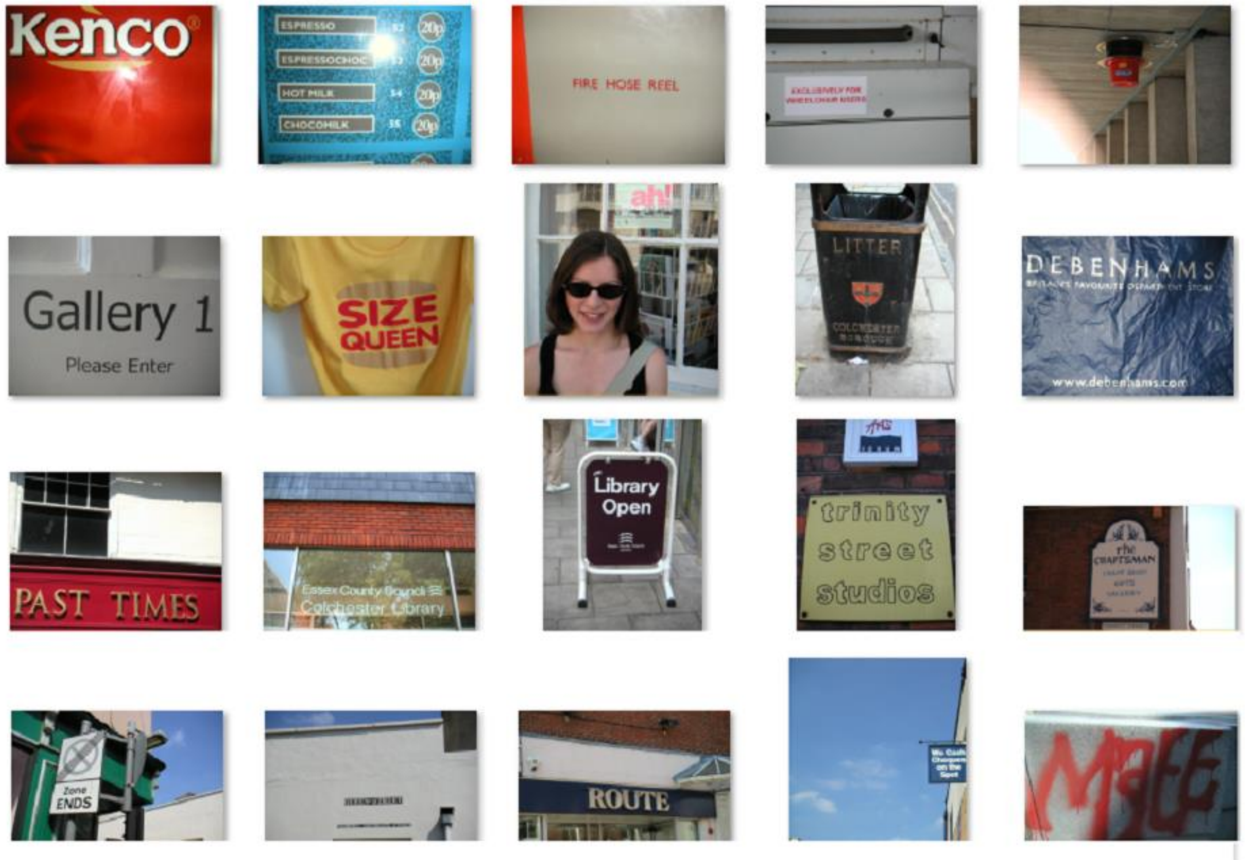


Figure 1: Example images for ICDAR Dataset

ICDAR 2003 and **ICDAR 2011** are the two most commonly used datasets.

How do we measure the performance?
The performance is measured by **Precision**, **Recall** and **F-measure.**

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

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

$$f = \frac{1}{\frac{\alpha}{p} + \frac{1-\alpha}{r}}$$

T: ground-truth set of targets
E: the set returned by the system under test, called estimates.
$\mathbf{m_p}$: match between two rectangles
m(r, R): The best match for a rectangle ***r*** in a set of rectangles **R**.

$$m(r, R) = \max m_p(r, r') || r' \in R$$

Challenges

Why is it more challenging, compared to traditional OCR?

1. Text Variations:

- Pattern
- Font
- Color
- Scale
- Orientation

2. Background Complexity

- Cluttered background
- Complex background that resembles text

3. Difficulties Introduced by Camera

- Uneven lighting
- Illumination
- Blur
- Low resolution
- Perspective distortion

Recent Works

What are the major approaches in previous works?

1. Region Based Approaches

This kind of approaches focus on efficient binary classification of small image patches, often in a sliding window scheme.

Drawbacks:
Unknown knowledge of text properties such as scales, colors, and orientations are quite challenging for accurate and robust classification.

2. Component Based Approaches

In these approaches, connected components are extracted first, and then non-text components are pruned based on heuristic rules or with trained classifiers.

Drawbacks:
(1)The construction of component is sensitive to image noise and distortion;
(2) the subsequent filters or classifiers may not be effective enough for removing non-text components.

Flowchart of the Proposed Algorithm

What is the proposed pipeline?

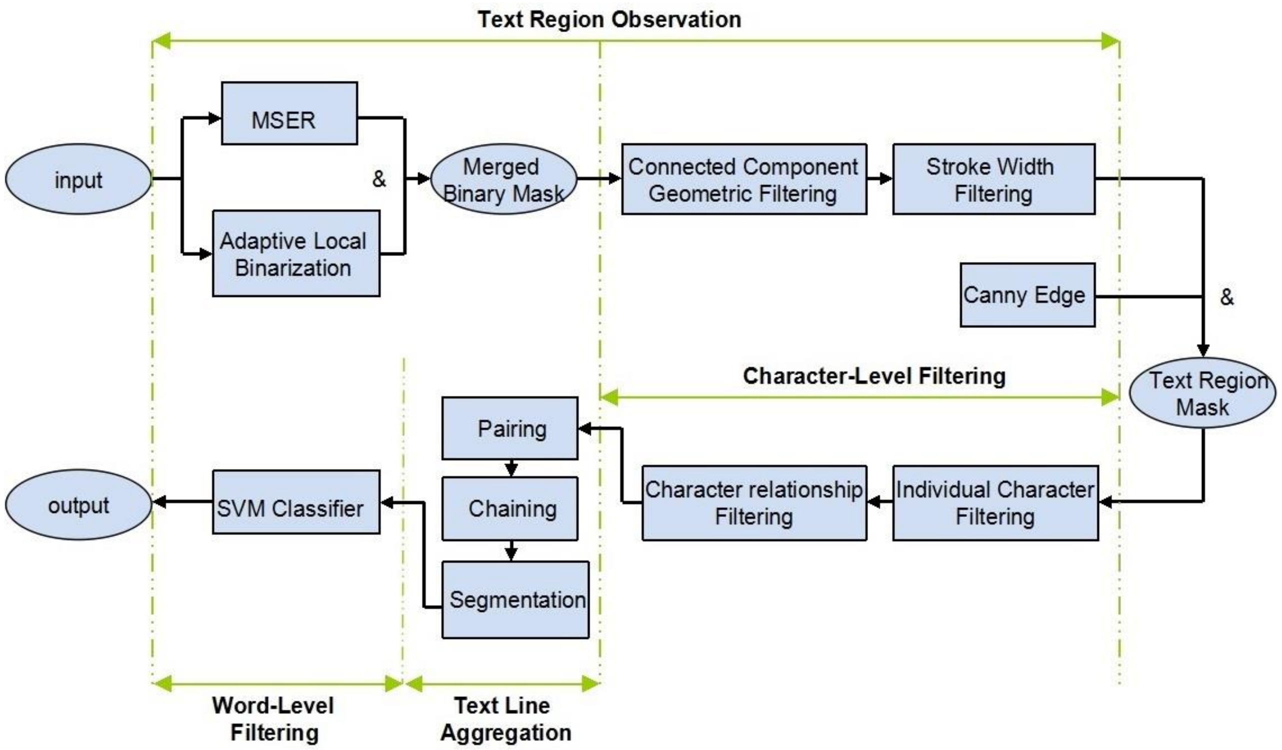


Figure 2: Flowchart of the proposed algorithm

Our proposed method aims to combine the advantages of both region-based and component-based methods, while overcoming their inherent limitations.

How do we incorporate the two kinds of methods?

1. 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.

2. We then develop a **region-based method** to filter out the vast amount of non-text components. Without a sliding window or multi-scale scanning, the proposed method is much less computationally expensive compared to existing region-based methods.

MSER for Text Region Detection

Maximally **S**table **E**xtremal **R**egions (MSER)

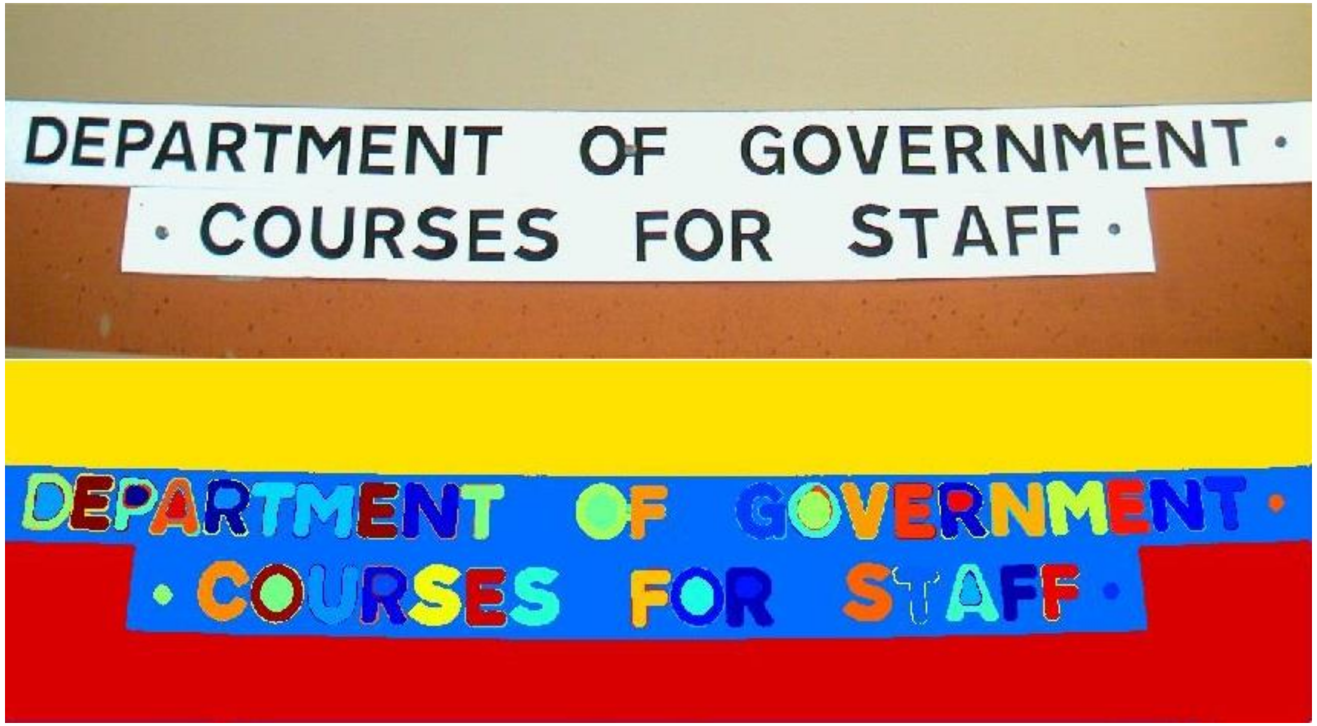


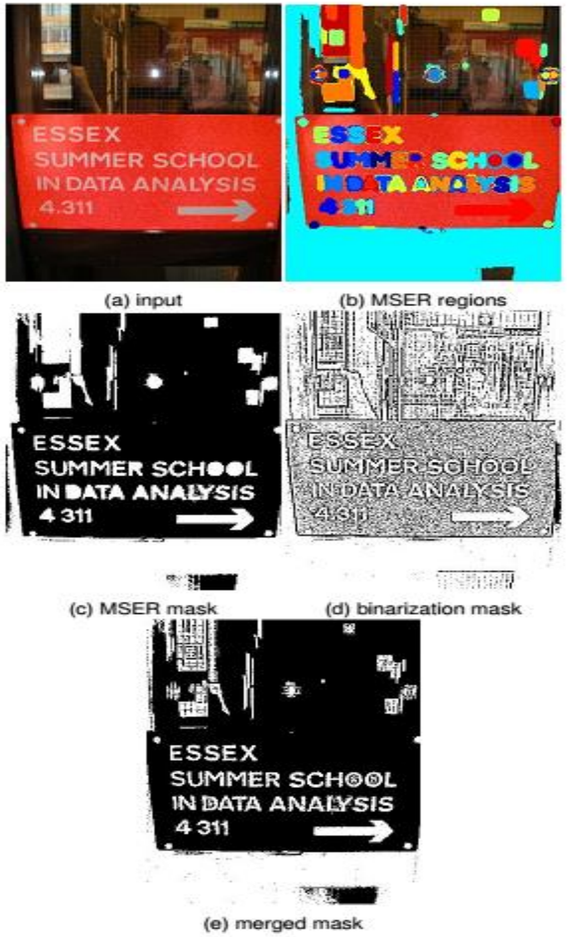
Figure 3: MSER extraction of CCs

- MSER is often used as a **C**onnected **C**omponent (CC) extractor. However, here we use it as a detector instead, whose task is to determine the candidate text image regions, represented by a **binary mask**.

  ❖ During our experiments, we purposefully lower down the threshold during MSER analysis to **maximize the detection probability of text regions**.

Local Adaptive Binarization

A complementary text region extractor.



(a) input

(b) MSER regions

(c) MSER mask

(d) binarization mask

(e) merged mask

Figure 4: Text region extraction

- MSER is sensitive to image blur, so we use Local Adaptive Binarization as a complementary approach for text region extraction.

  What common Binarization methods exist?
  1. Traditional binarization methods based on global thresholding are unable to achieve efficient binarization for different image contents.
  2. Local adaptive binarization methods, on the other hand, examine local statistics in the neighborhood and determine the binarization threshold at the pixel level.

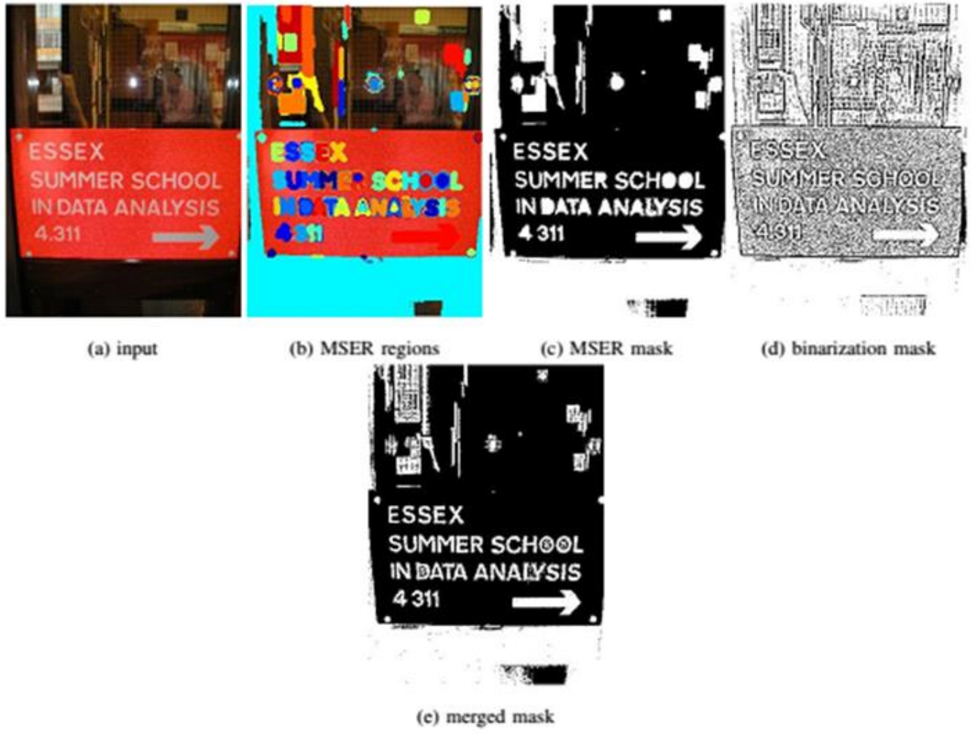
  Why chose Local Adaptive Binarization?
  1. It works well on both small and large text region extraction.
  2. It is robust to illumination change.

□ 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.

MSER and Local Adaptive Binarization

Why do we use two Text Region Extractors?

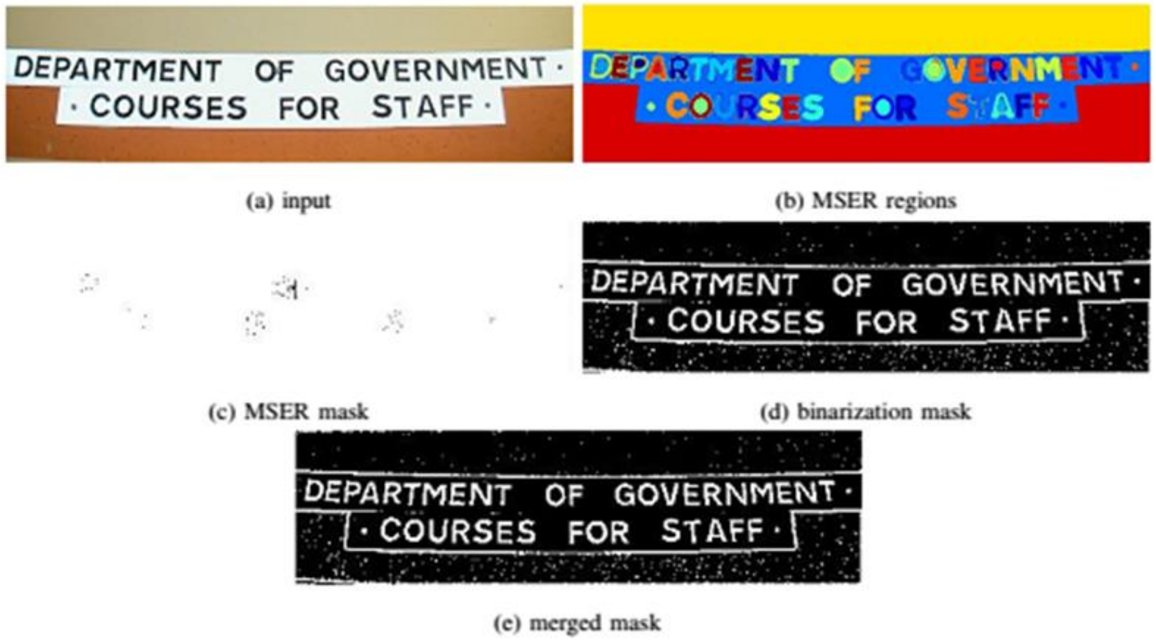
❑ From the following figures, we can observe in different situations that MSER and Local adaptive binarization are two methods that are complementary to each other. Fig.5 shows that **MSER** detects the text region better under some circumstance, while Fig.6 shows that **Local Adaptive Binarization** method extracts the text region better in another situation. Therefore, proper combination of the two masks is essential.



(a) input (b) MSER regions (c) MSER mask (d) binarization mask

(e) merged mask

Figure 5: The MSER method contributes to the extraction of text region while the local adaptive binarization method helps with clearing the edge. The merged mask renders better text region than either method alone.



(a) input (b) MSER regions

(c) MSER mask (d) binarization mask

(e) merged mask

Figure 6: The MSER method over-extracts the regions, while the adaptive binarization mask renders more specific regions.

Geometric Filtering&Stroke Width Transform

Character-Level Filtering on Text Candidate Regions

- After the merging of text regions, we apply Connected Component Analysis(**CCA**) to extract text components. At this stage, each component corresponds to a character.

What properties do we use to filter character candidates against non-text CCs?

- **Basic properties** of characters, such as

  1. **area**
  2. **eccentricity**
  3. **solidity**

  are used to filter out non-character components.

- To further remove non-character components, we also use the **stroke width properties** of text.

  The consistency of stroke width among texts, is the interior property of texts.

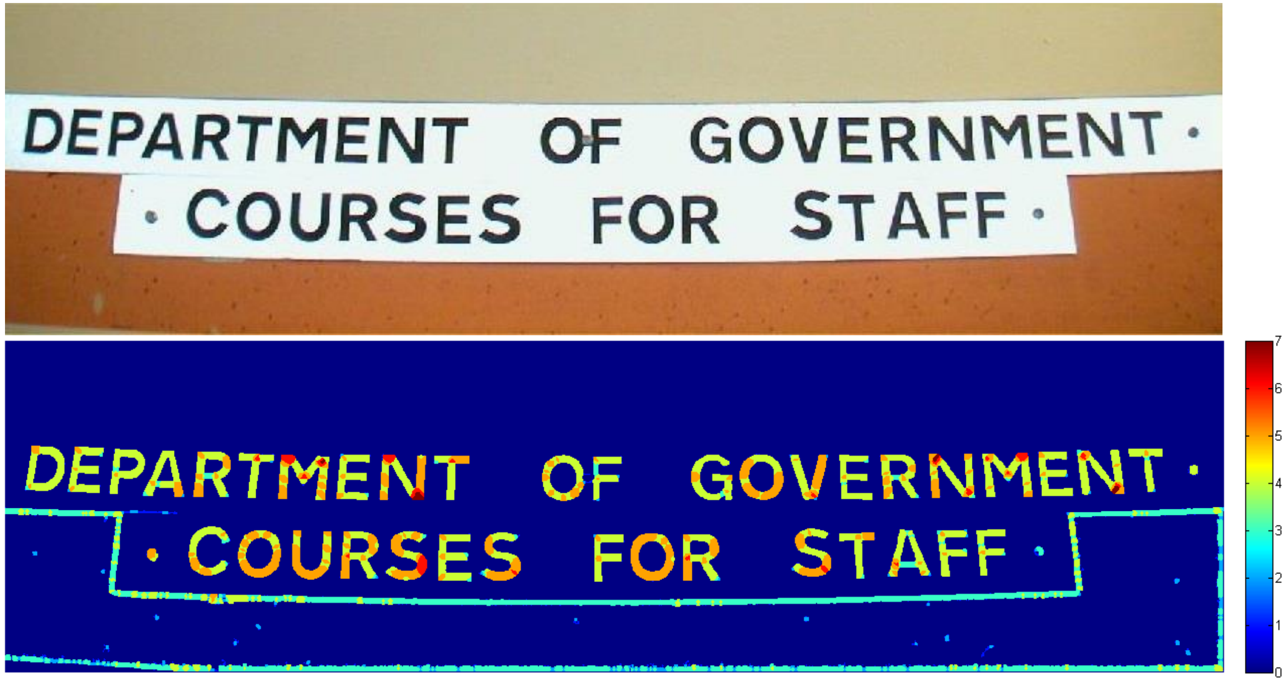


Figure 7: Stroke widths among texts are consistent

Edge Information

Is edge information of texts helpful?

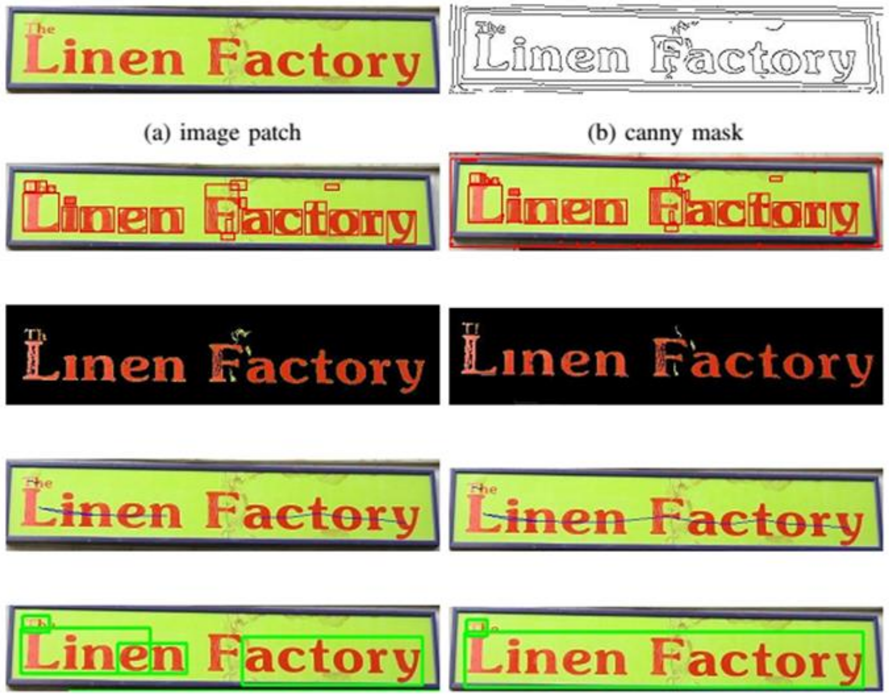


Figure 8: The comparison between two methods, one using edge information

❑ YES.

Why?

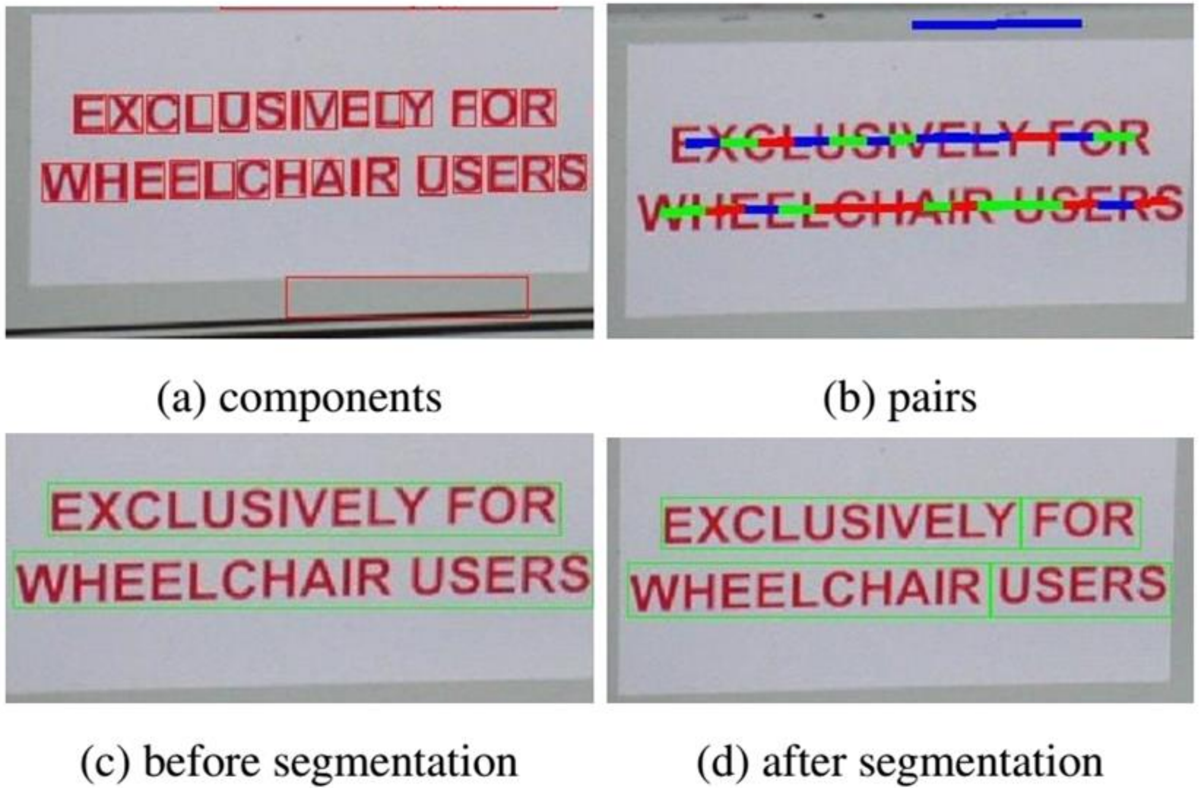
Edge information helps prevent aggregating text components with adjacent noise components, thus improving the performance of text word detection performance.

Example

- On the left column, the images illustrate the process of text word aggregation without using canny edge mask.
- On the right column, the images justifies the necessity of edge information.

Text Line Aggregation

The preparation for Word-Level Filtering



(a) components

(b) pairs

(c) before segmentation

(d) after segmentation

Figure 9: The text line aggregation process

□ Once we have the filtered text region mask, we apply CCA again to obtain <u>comparatively pure</u> text components and apply text line aggregation to derive text lines.

Why does it work?

- 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 <u>efficiently filter out some non-text components</u>.

How about multiple words in a single line?

- 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 <u>segment word candidates</u> when necessary.

Word-Level Filtering

How does word-level filtering work? Why?

- In the previous steps, individual components are grouped and linked into word candidates.

We propose to **train a classifier** to determine if a word candidate (an image region) is text or not.

Why does it work?

1. It operates 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.
2. 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, it involves much less computational complexity.

To characterize the word image regions, we use **HOG** (histogram of oriented gradients) and **LBP** (local binary pattern) features.

Why these feature?

1. HOG feature is efficient in capturing the edge and shape information of text and differentiating them from the background.
2. LBP feature is efficient in capturing the difference of texture characteristics between the text and background.

How are these features used?
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.

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).

Results:

Detection Examples for ICDAR2003 and ICDAR 2011

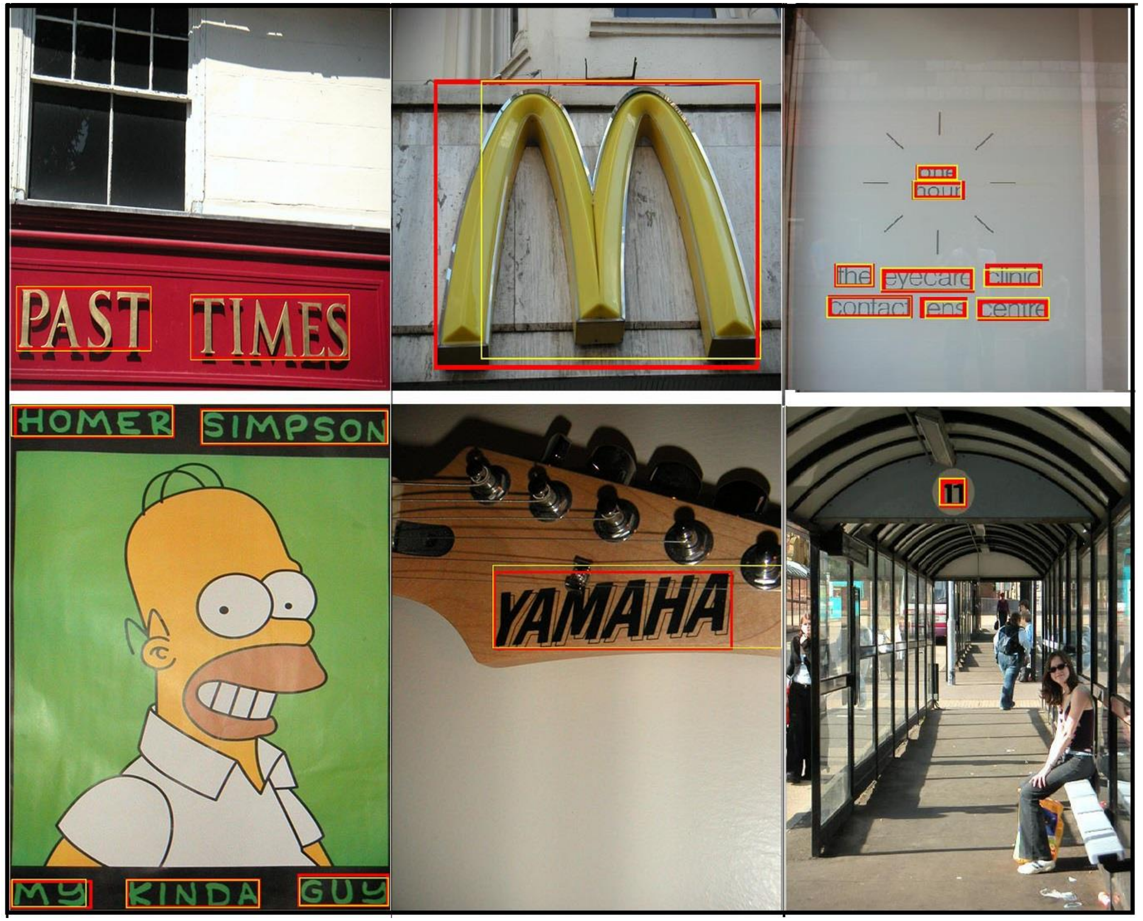
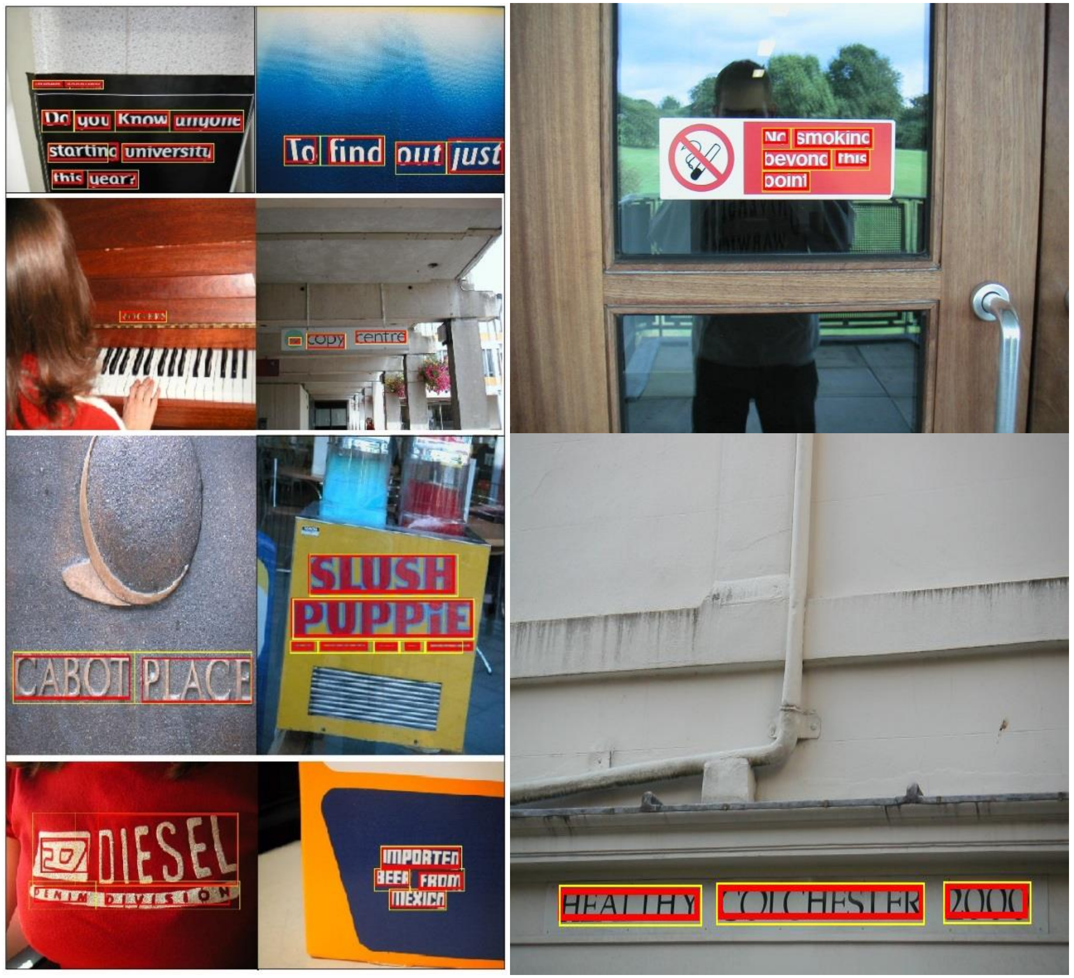


Figure 10: Scene text detection result examples: Bounding rectangles of the ground truth are indicated in red while the detection results in yellow.



Results:

Incorrectly Labeled Image Examples, and Why?

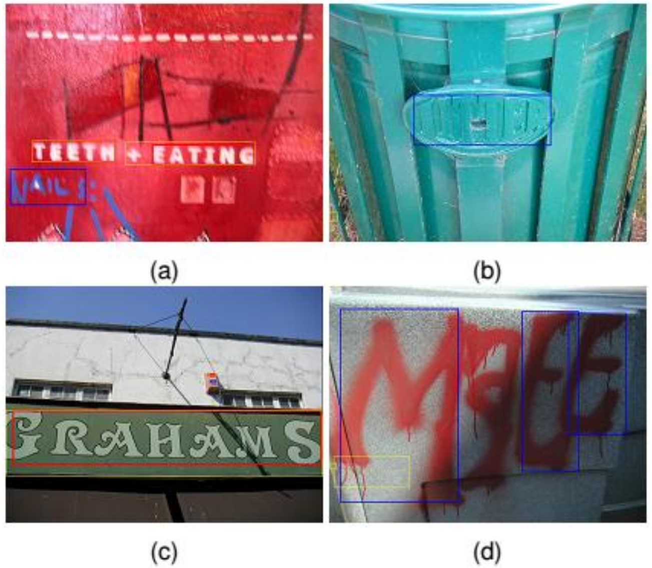


Fig. 11: Incorrectly labeled image examples in ICDAR2003 dataset. These cases include: hand-written characters that do not survive the SVM classifier (a), subtle texts that resemble background too much (b), extremely small text that are suppressed as noise by geometrical filtering (c), Text components that does not pair (d). The ground truth is indicated by red bounding boxes while the detection results in yellow. Blue rectangles indicate ground truth that has not been detected.

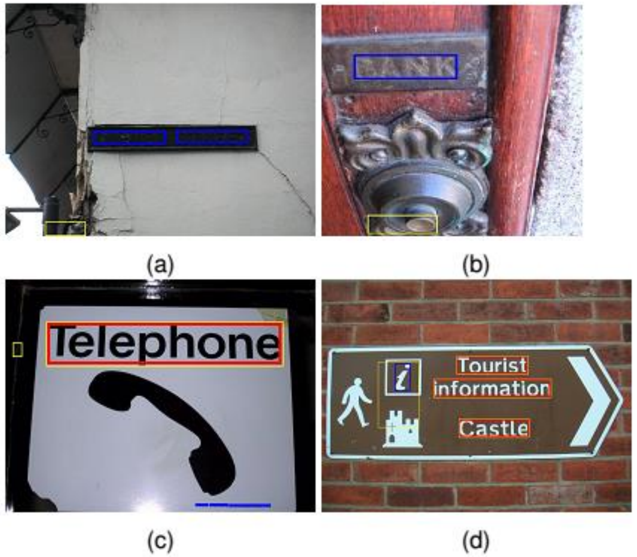


Fig. 12: Incorrectly labeled image examples in ICDAR2011 dataset. These cases include: characters that are not maximally stable and therefore not regarded as candidates the (a), subtle texts that resemble background too much (b), extremely small text that are suppressed as noise by geometrical filtering (c), Text components that pair to text-like graphs (d). Likewise, the ground truth is indicated by red bounding boxes while the detection results in yellow. Blue rectangles indicate ground truth that has not been detected.

Results:

Comparing with Other Methods

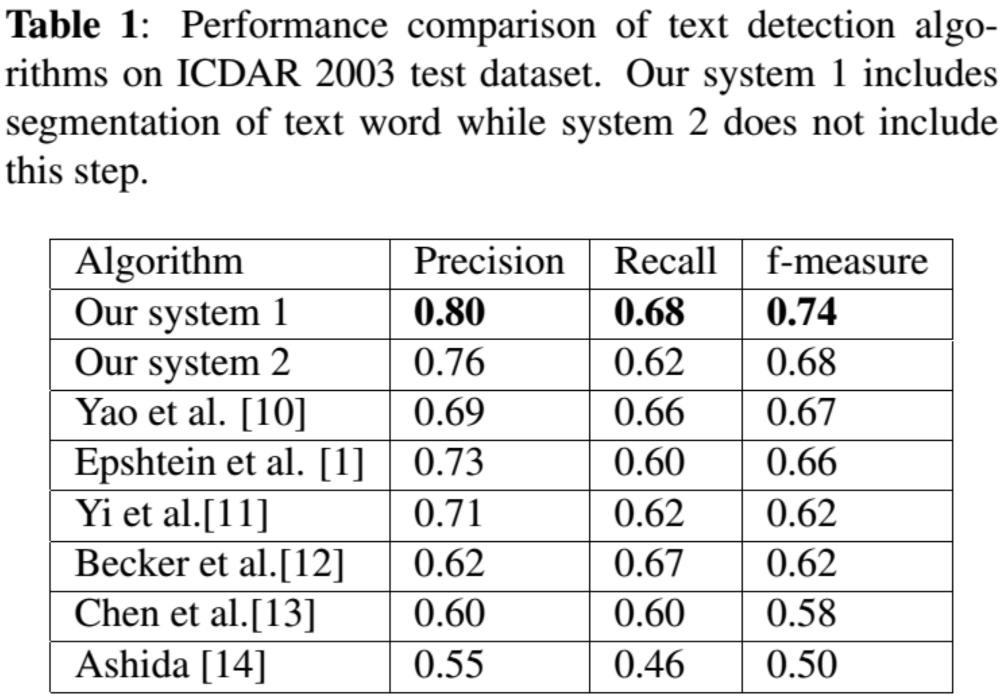


Table 1: Performance comparison of text detection algorithms on ICDAR 2003 test dataset. Our system 1 includes segmentation of text word while system 2 does not include this step.

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

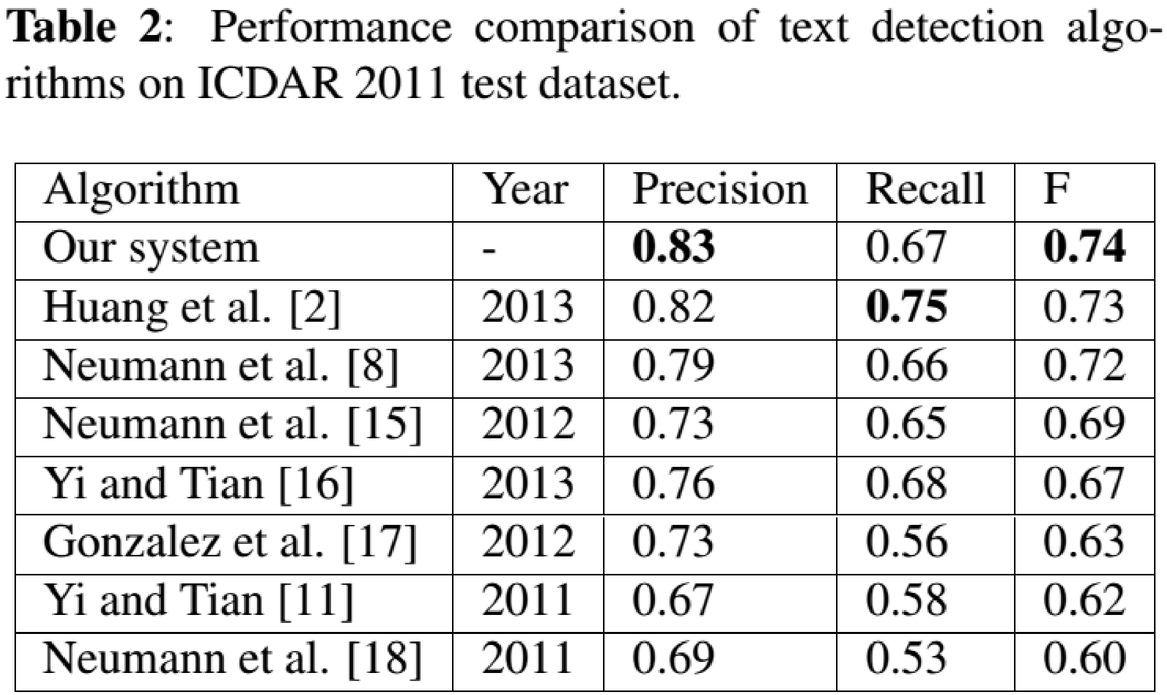


Table 2: Performance comparison of text detection algorithms on ICDAR 2011 test dataset.

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

Results:

Comparing with Various Parameters

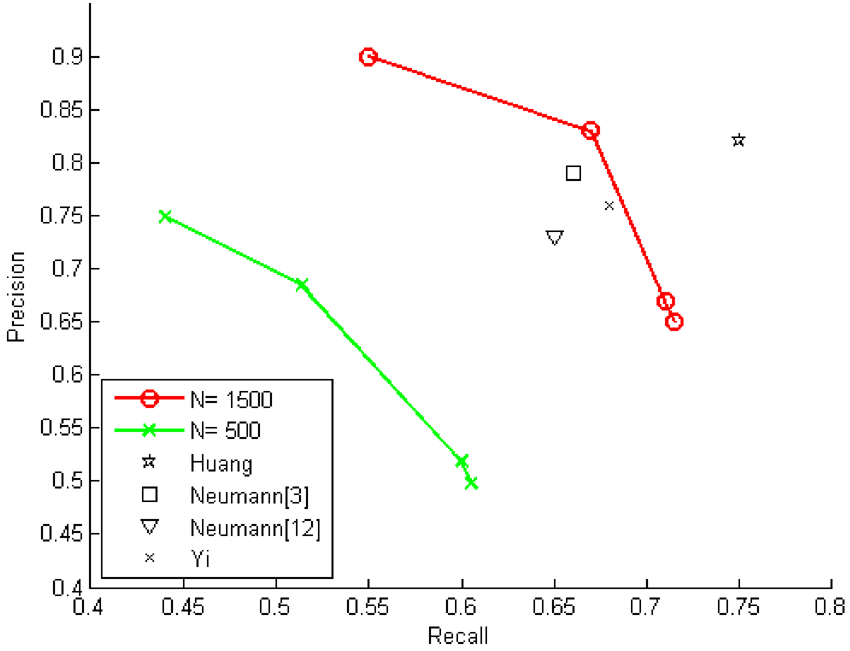


Figure 13: The precision-recall curves with different MSER parameters and various codebook sizes.

How do we obtain a precision-recall curve?

- 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.

Why multiple precision-recall curves?

- Different choices of N.

  When the codebook size N is too small, the codebook does not summarize the feature well, therefore, resulting in low precision and recall.

Conclusion

And contributions of this paper

- We developed a novel scene text detection algorithm that couples component-based with region-based 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 <u>major contribution</u> of this paper lies in the following two major aspects.

1) Our algorithm combines the advantages of component-based and region-based methods for text detection.
2) We couple text component detection and word-level patch classification to achieve highly efficient text detection.