Maximally Stable Extremal Regions and Naïve Bayes to Detect Scene Text

E Rainarli
Department of Informatics Engineering, Universitas Komputer Indonesia, Jl Dipatiukur 112 - 116 Bandung, Indonesia

Email: ednawati.rainarli@email.unikom.ac.id

Abstract. This study examines the performance of Maximally Stable Extremal Regions (MSER) and Naïve Bayes in detecting scene text. The variance of types and sizes of fonts, uneven lighting conditions, the text orientation, a complex background, occlusion, and the presence of objects that resemble text, make the scene text detection is quite challenging. The initial stage of the detection process is to use MSER to get the candidate characters in the image. The validation process of the candidate character uses the Naïve Bayes classifier, which we trained using char74k and CIFAR10 data sets. The classification process used HOG as the extracted features. The system validates the candidates by comparing the Naïve Bayes probability value with the specified threshold value. By using 100 images from ICDAR 2015, the research obtains a 50% reduction of the candidate with an accuracy increase of 8% for Naïve Bayes using threshold values. The result shows that Naïve Bayes with the thresholding value is better than the usual Naïve Bayes classification in selecting candidates.

1. Introduction

Research on scenes text detection and recognition is developing rapidly along with the development of computer vision. The flexibility of using a mobile phone to take pictures, makes it possible to use input in the form of images to obtain information. Machine translation, image retrieval, information extraction are some implementations of scene text recognition. There are two approaches to generating text candidates, namely using Region-Based and Connected Component-Based [1]. MSER is one of the popular methods that use the Connected Components-Based approach. This method applies the watershed principle to segment candidate objects that will be recognized [2]. The advantages of this method are resistant to scaling and rotation. That advantages are suitable for following the text scene that is varying in size and multi-orientation. MSER performs segmentation by determining areas that have a stable connected component of some gray-level sets of the image [3].

Neumann [3] was the first to use the MSER method to detect text. There were some stages. It started with character classification using SVM and continued with forming text-lines to get bounding boxes, and the geometric normalization to detect the non-horizontal text. Turkey [4] combined MSER with CNN to extract features while Huang [5] approached MSER with the sliding window. He [6] focused on improving MSER by adding a contrast enhancement process. Both Huang and He used CNN to detect the text. Another researcher, Soni [7], exploited the edge smoothing process before doing MSER and did not use the deep learning method to get character candidates. Candidates for the obtained characters are then selected using rules or learning classifier. The rules used to select text include aspects of ratio, stroke width, color consistency, compactness, and other possible features [8]. Support Vector Machine [3,9,10], Naïve Bayes [11], Ada Boost [1], and Random Forest [12] methods are some methods to classify the candidate regions, as characters or not characters. This research used Naïve Bayes to
determine the areas of a font. We use a threshold of the probabilistic value to select text candidates as a font. Usually, Naïve Bayes Classifier compares the probabilistic value between the two classes. We examine some probabilistic values, for example, 0.75 or 0.9, to get a higher precision value. Therefore, in this study, MSER and Naïve Bayes will be used to select candidate characters in the scene text detection. This research would examine the performance of Naïve Bayes to detect text using determined threshold value. We used precision, recall and f-measure to measure the performance of Naïve Bayes.

2. Method
There were two stages for detecting characters with MSER and Naïve Bayes classification, namely training classifiers and the detection process. The top part of Figure 1 is the Naïve Bayes classifier training process, while the bottom part is the text detection process. In Figure 1, the first process was resizing and grayscale from the Char74K [13], CIFAR10 [14] data set. Data was set to detect candidate characters is Char74K, consisting of 12,500 images of numbers, lowercase letters, and capital letters with various letter sizes and image sizes. We used a CIFAR10 data set, consist of 60,000 images from 10 types of objects with an image size of 32x32, to detect non-text. The process continued with down sampling for non-texted data so that the numbers were as large as 12500. From this data set, we split into training data and validation data with a ratio of 2:1. Modeling began with resizing and grayscale. We continued the process with feature extraction, using the Histogram of Gradient (HoG). Finally, we used the extraction results to train and validate the Naïve Bayes model.

The second stage was the process of detecting text scenes. We used a sample of 100 images from the ICDAR 2015 data set [15]. At the bottom of Figure 1, the initial process was grayscale and used MSER to get character candidates. The next process was extracting the HOG feature and validating candidate characters using a trained classifier. To reduce positive false, we filtered the candidate characters based on their probabilities. We selected the classification results using the area of text box and aspect ratio to reduce false positives. The final stage was the merging of intersecting boxes using Non-Maximum Suppression (NMS). The output of this process was the character bounding box. We compared the precision of the output using the original Naïve Bayes and the Naïve Bayes using thresholds.

3. Results and Discussion
3.1. Model of Classifier
We measured the performance of the Naïve Bayes classifier model using validation data for non-text and text data, respectively 4117 and 4113. The average of precision, recall, and F-measure reached 0.88. Table 1 presents details of the results. We used the parameter values from the Naïve Bayes validation result to select text candidates from the MSER extraction. Based on [16], it explains that Naïve Bayes still worked effectively, even though we have not checked the assumptions independent of the features. The high-performance value, in this case, reinforces the notion that the data set of text and non-text were also linear, as appropriate as stated by Rish [16]. Table 1 shows the high precision values in recognizing
objects like a letter, in which the value achieved 0.90. Based on this result, we conclude that this Naïve Bayes classifier has a potential model in text scene detection.

|               | Precision | Recall | F-Measure | Number of Data |
|---------------|-----------|--------|-----------|----------------|
| Non-Text      | 0.86      | 0.91   | 0.88      | 4117           |
| Text          | 0.90      | 0.85   | 0.88      | 4133           |
| Average       | 0.88      | 0.88   | 0.88      |                 |

3.2. MSER and The Classification of Candidate Character
The scene text detection process used the ICDAR 2015 data set. It used MSER to generate candidate scene texts from grayscale images. Figure 2 is one example of detecting text candidates' process. We continued the process with resizing, grayscale, and HOG feature extraction. The process examined the candidate text using a threshold value that was determined by the researcher. In this study, it chose 0.1 as a threshold value. In Figure 2, the blue boxes are the result of detection from MSER. Some of the detected boxes are not text. The result is consistent with that presented by Soni [11]. MSER produces a great candidate but need to do a filter to obtain the candidate character. Figure 3 shows the classification results. In Figure 4, it filters the letters based on the aspect ratio rules and candidate area. In Figure 5, there are still errors in detecting scene text. The process removes the non-text boxes.

![Figure 2. Candidate Text Box of MSER.](image)

![Figure 3. Naïve Bayes Selection with Threshold 0.1](image)
3.3. False Positive Reduction and NMS

Text candidates from the Naïve Bayes selection still contained images that are not text. For this reason, the selection results of Naïve Bayes were re-selected using rule-based, namely aspect ratio [3] and area [17]. The final stage was to supervise intersecting text boxes using the NMS process, which suppressed the text box with an intersection > 0.8. Figure 5 shows the final result of scene text detection. The use of rule-based is to eliminate false-positives. The process deleted the dense text that recognized as a character at a previous stage. Based on Figure 2, we concluded that it is difficult using a character-level approach in detecting dense text. This process removes several text boxes. Figure 6 explains the details of the algorithm for detecting box selection. Line 13 from Figure 6 shows the algorithm delete box in which the candidate area is too small (less than 30) or too large (more than 4500). From line 17 of Figure 6, we rejected the aspect ratio that more than two or less than 2.5. All values are obtained from the results of the experiment.

```
1: function selection_box (result_boxes, gray_images, loaded_model):
2:    b_box = prepare list of selection result
3:    for i = 1 to length(result_boxes) do
4:        x1, y1, x2, y2 = result[i]
5:        crop_image = gray_images[y1:y2, x1:x2]
6:        crop_resize = resizing (crop_image)
7:        img_hog = extract_feature (crop_size)
```
8: probability_value = predict_probability(img_hog, loaded_model)
9: if probability_value(nontext) > 0.1 then
10: continue
11: end if
12: w=x2-x1, h=y2-y1, area = w*h
13: if (area < 30) or (area > 4500):
14: continue
15: end if
16: aspect_ratio = w/h
17: if (aspect_ratio > 2) or (aspect_ratio <0.25):
18: continue
19: end if
20: b_box[i]=x1,y1,x2,y2
21: end for
22: return b_box
23: end function

**Figure 6.** The Algorithm of candidate character selection

### 3.4 Detection Result

Table 2 shows a comparison of the precision of using threshold values with classifications. The number of correctly detected candidate characters does not change. The difference lies in fewer false-positive characters. Nevertheless, the accuracy value has not produced a high value. We used 100 images from the 2015 ICDAR data set. The Naïve Bayes with a threshold able to reduce the number of detecting texts, half of the Naïve Bayes Classification. Although the number of correct text boxes is 877 and it is less than Naïve Bayes Classification. The precision is more than Naïve Bayes classification, which is 0.2385. The use of the threshold is similar to Iqbal [18], which uses threshold in the Bayesian Network to select candidates.

|                   | Naïve Bayes With Threshold | Naïve Bayes Classification |
|-------------------|-----------------------------|-----------------------------|
| Number of True Text Box | 877                         | 1117                        |
| Number of Detection | 3677                        | 7100                        |
| Precision         | 0.2385                      | 0.1573                      |

During the detection process, there are several conditions causing false-positive. Those examples of scene text that were successfully detected. The dark condition in Figure 7, the complexity of the background from Figure 8, the presence of similar objects in Figure 10, such as windows, fences, lights are some of the results of the proposed scene text detection. Figure 9 is the example of using MSER we able to detect text with multi-orientation.

**Figure 7.** The Dark Condition
Figure 8. The Complexity of the Background

Figure 9. The Example of Using MSER

Figure 10. The Results of the Proposed Scene Text Detection
4. Conclusion
Based on the test results, it has been shown that the use of threshold value on Naïve Bayes can reduce the false positive value in the scene text detection. Even so, the resulting accuracy is still low. Detection using character levels has constraints when detecting dense text. For the next research, the researcher considers using a word level approach. Selection of extraction features that not only use low-level features, optimizing threshold values and using the different datasets to create a classifier are several ways to produce more accurate text scene detection.

References
[1] Yin, X., Yin, X. C., Hao, H. W., & Iqbal, K. 2012. Effective text localization in natural scene images with MSER, geometry-based grouping and AdaBoost. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012) (pp. 725-728). IEEE.
[2] Matas, J., Chum, O., Urban, M., & Pajdla, T. 2004. Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing, 22(10), pp. 761-767.
[3] Neumann, L., & Matas, J. 2010. A method for text localization and recognition in real-world images. In Asian Conference on Computer Vision (pp. 770-783). Springer, Berlin, Heidelberg.
[4] Turki, H., Halima, M. B., & Alimi, A. M. 2017. Text detection based on MSER and CNN features. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) 1 pp. 949-954. IEEE.
[5] Huang, W., Qiao, Y., & Tang, X. 2014. Robust scene text detection with convolution neural network induced msr trees. In European conference on computer vision (pp. 497-511). Springer, Cham.
[6] He, T., Huang, W., Qiao, Y., & Yao, J. 2016. Text-attentional convolutional neural network for scene text detection. IEEE transactions on image processing, 25(6), pp. 2529-2541.
[7] Soni, R., Kumar, B., & Chand, S. 2018. Extracting Text Regions from Scene Images using Weighted Median Filter and MSER. In 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) (pp. 915-920). IEEE.
[8] Zhu, Y., Yao, C., & Bai, X. 2016. Scene text detection and recognition: Recent advances and future trends. Frontiers of Computer Science, 10(1), pp. 19-36.
[9] Chen, J., & Takagi, N. 2016. Gray-scale morphology-based image segmentation and character extraction using SVM. In 2016 IEEE 46th International Symposium on Multiple-Valued Logic (ISMVL) (pp. 177-182). IEEE.
[10] Huang, X., Shen, T., Wang, R., & Gao, C. 2015. Text detection and recognition in natural scene images. In 2015 International Conference on Estimation, Detection and Information Fusion (ICEDIF) (pp. 44-49). IEEE.
[11] Soni, R., Kumar, B., & Chand, S. 2017. Text detection and localization in natural scene images using MSER and fast guided filter. In 2017 Fourth International Conference on Image Information Processing (ICIIP) (pp. 1-6). IEEE.
[12] Huang, W., Lin, Z., Yang, J., & Wang, J. 2013. Text localization in natural images using stroke feature transform and text covariance descriptors. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1241-1248).
[13] De Campos, T. E., Babu, B. R., & Varma, M. 2009. Character recognition in natural images. VISAPP (2), p. 7.
[14] Krizhevsky, A., & Hinton, G. 2009. Learning multiple layers of features from tiny images.
[15] Zhou, X., Zhou, S., Yao, C., Cao, Z., & Yin, Q. 2015. Icdar 2015 text reading in the wild competition. arXiv preprint arXiv:1506.03184.
[16] Rish, I. 2001. An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence 3(22), pp. 41-46.
[17] Yin, X. C., Yin, X., Huang, K., & Hao, H. W. 2013. Robust text detection in natural scene images. *IEEE transactions on pattern analysis and machine intelligence, 36*(5), pp. 970-983.

[18] Iqbal, K., Yin, X. C., Hao, H. W., Asghar, S., & Ali, H. 2014. Bayesian network scores based text localization in scene images. In *2014 International Joint Conference on Neural Networks (IJCNN)* (pp. 2218-2225). IEEE.