Scene Text Detection with Robust Character Candidate Extraction Method

Myung-Chul Sung[1], Bongjin Jun[2], Hojin Cho[2] and Daijin Kim[1]

[1]*Department of Computer Science and Engineering, POSTECH, Pohang, Korea*
[2]*StradVision Incorporation, Dellaware, USA*
Email: mcs8779@postech.ac.kr, bongjin.jun@stradvision.com, hojin.cho@stradvision.com, dkim@postech.ac.kr

***Abstract*—The maximally stable extremal region (MSER) method has been widely used to extract character candidates, but because of its requirement for maximum stability, high text detection performance is difficult to obtain. To overcome this problem, we propose a robust character candidate extraction method that performs ER tree construction, sub-path partitioning, sub-path pruning, and character candidate selection sequentially. Then, we use the AdaBoost trained character classifier to verify the extracted character candidates. Then, we use heuristics to refine the classified character candidates and group the refined character candidates into text regions according to their geometric adjacency and color similarity. We also apply the proposed text detection method to two different color channels C_r and C_b and obtain the final detection result by combining the detection results on the three different channels. The proposed text detection method on ICDAR 2013 dataset achieved 8%, 1%, and 4% improvements in recall rate, precision rate and f-score, respectively, compared to the state-of-the-art methods.**

I. Introduction

Text regions in images of natural scenes include important information for content-based image analysis such as web image search and video information retrieval [1]–[3]. However, text regions in natural scene images usually vary greatly in font, size, color, appearance and illumination, and can have complex backgrounds that differ from those in documents and business cards. These variations make text detection difficult in natural scene images.

Existing methods for text detection can be largely categorized into three groups: methods using the sliding window [4]–[6], methods using the connected components [7]–[9], and hybrid methods [10].

Neumann and Matas [11] used maximally stable extremal region (MSER) [12] to detect text. Chen et al. [13] combined Canny edges with MSER to overcome MSERs sensitivity to blurred images and to detect small text regions even in limited resolutions. They removed the pixels outside boundaries that were formed by the Canny edges. To incorporate geometric information about the text regions, Shi et al. [14] used graph models that were built on the MSERs. To remove the repeating ERs, Yin et al. [15] used various algorithms that could prune the MSER results. Many other papers [16]–[18] have also used MSERs to extract character candidates and have reported promising text detection results on widely used datasets. The main advantage of using MSER for character candidate extraction is that it can detect text regions regardless of their scale and is robust to noise, and to affine illumination variations.

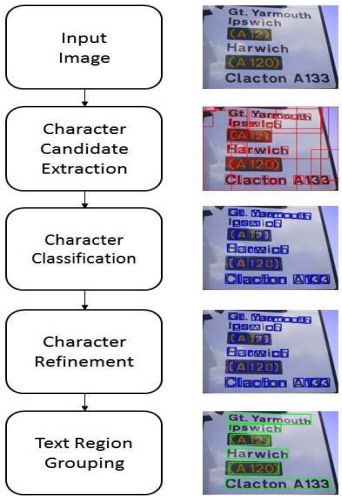

Fig. 1. Overall process of the proposed text detection method.

However, we obtained some incorrect extraction results when we used MSER to extract character candidates. To overcome this problem, we propose a novel method (Fig. 1) to extract character candidates from an ER tree. We divided local paths in an ER tree into multiple sub-paths according to the size and position similarities of ER regions in the path. Then we pruned unnecessary sub-paths; some sub-paths with high probability of containing characters remained. Then from each of the remaining sub-paths we selected the ER that had the minimum regularized stability value as character candidates.

After extracting character candidates, we used an AdaBoost-trained classifier that uses mean local binary patterns (MLBP) [19] to classify each character candidate into either character or non-character. Finally, we use simple heuristic rules to filter out misclassified characters, and group the remaining characters into text regions. Fig. 1 shows overall process of the proposed text detection method.

II. Character Candidate Extraction

The process of the proposed candidate extraction method consists of four steps: ER tree construction, sub-path partitioning, sub-path pruning and character candidate selection.

A. ER Tree Construction

For a gray-scale image I, a binary threshold image $B_t(p)$ can be obtained at a threshold level t by thresholding as

$$B_t(p) = \begin{cases} 1 & \text{if} \quad I(p) \geq t, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where $t \in [min(I(p)),\ max(I(p))]$ and p denotes a specific position in I.

An extremal region R_t at a threshold level t is defined by a connected component in the binary image $B_t(p)$ as

$$\forall p \in R_t,\ \forall q \in boundary(R_t) \quad \rightarrow \quad B_t(p) \geq B_t(q), \tag{2}$$

where p and q are position indexes. Extremal regions can be managed and implemented efficiently by using an ER tree [20], in which each node represents a connected component (or extremal region R_t).

B. Sub-Path Partitioning

Many approaches to text detection have used MSER [12] to extract character candidates. An MSER is defined as the extremal region that has the smallest stability value Ψ among all extremal regions R_t along its local path:

$$\Psi(R_t) = \frac{(|R_{t-\Delta}| - |R_{t+\Delta}|)}{|R_t|}, \tag{3}$$

where $|R_t|$ denotes the cardinality of R_t and Δ can be chosen appropriately according to different applications.

However, a considerable amount of character regions were not extracted when using MSER to extract character candidates. The MSER method selects only one ER (from each local path) that has the minimum stability value (Ψ) in its local path in the ER tree; the selected ER may not be a character region. Fig. 2(a) shows a part of an ER tree and the resulting extracted regions (bounding boxes) using MSER. The maximally stable region selected from the two local paths shown are the outermost bounding boxes and they do not correspond to a character.

This non-selection problem of character regions occurs when the stability values of extremal regions with quite different geometric properties (i.e. area and position in image) are compared along a local path in the ER tree. For example, comparing stability values of ERs at threshold level 185 (R_{185}) to those at threshold level 170 (R_{170}) is invalid because they are completely different in size and position (Fig. 2).

To address this problem, we propose a novel extraction method that divides local paths in an ER tree into multiple sub-paths according to the size and position similarities of ER regions. The similarity between two adjacent ERs is computed as

$$S(R_t, R_{t+1}) = \frac{A(R_t) \cap A(R_{t+1})}{A(R_t) \cup A(R_{t+1})}, \tag{4}$$

where $A(\bullet)$ denotes the bounding box of an ER R_t and t denotes the threshold level.

Then, we separate two adjacent ERs R_t and R_{t+1} in a local path into two different sub-paths (Fig. 2-b) when the similarity between R_t and R_{t+1} is smaller than a threshold value ϵ, which was empirically chosen as 0.7 in this work.

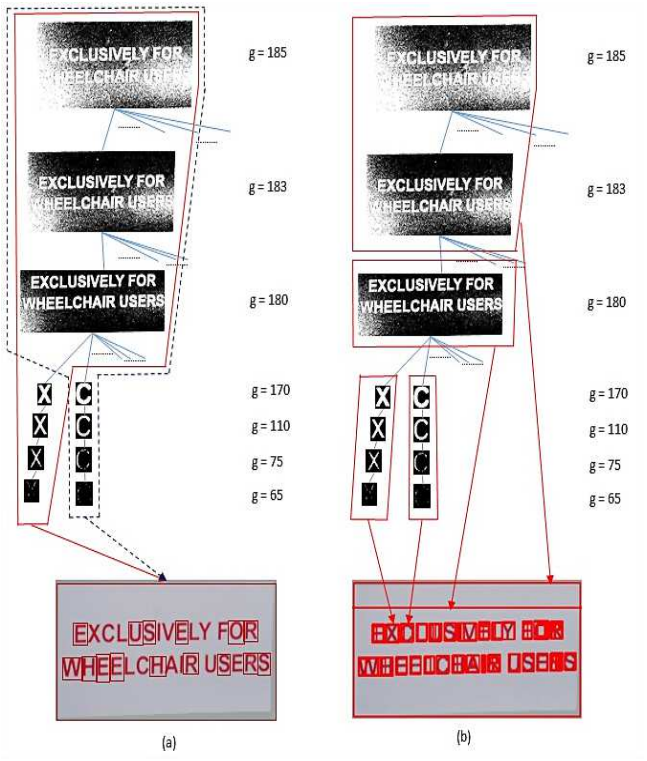

Fig. 2. Character candidates extracted from the ER tree: (a) the MSER extraction method, (b) the proposed extraction method. Bottom: bounding boxes around extracted character candidates.

Fig. 2(b) shows the same ER tree as 2(a), but the local paths have been divided into multiple sub-paths, resulting in 2 local paths being divided into more than 4 sub-paths. The results show that regions that could not be extracted when using the MSER method can be extracted when using the proposed extraction method.

C. Sub-Path Pruning

We observed that ERs are extracted over multiple threshold values if the image includes components with large intensity changes, such as edges or corners. Because the boundary of a character is edge-like, the ER sub-paths that contain characters are relatively longer than sub-paths that contain non-characters. Using this observation, we pruned the sub-paths that have depth smaller than a threshold value, which is set to 7 because experimental analysis using the ICDAR 2011 and ICDAR 2013 training datasets shows that most sub-paths that contain characters have depth larger than 7.

D. Character Candidate Selection

Among the remaining ERs in a sub-path, we penalize abnormal regions whose aspect ratios are too large or too small; to do this we used a regularized stability value [15] defined as

$$\bar{\Psi}(R_t) = \begin{cases} \Psi(R_t) + \theta_1(a - a_{max}) & \text{if} \quad a > a_{max} \\ \Psi(R_t) + \theta_2(a_{min} - a) & \text{if} \quad a < a_{min} \\ \Psi(R_t) & \text{otherwise}, \end{cases} \tag{5}$$

where a is the aspect ratio of R_t, θ_1 and θ_2 are the penalty ratios when $a > a_{max}$ and $a < a_{min}$, respectively, a_{max} and

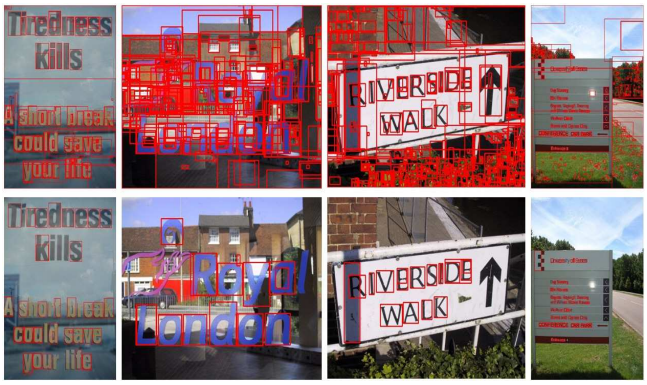

Fig. 3. Top row and bottom row: character candidates before and after character classification.

a_{min} are the maximum and minimum aspect ratios that an ER containing a single character is expected to have. In this work, we set θ_1 = 0.01, θ_2 = 0.35, a_{max} = 1.2, and a_{min} = 0.3 after experimental trials. Then we selected the ER with the smallest regularized stability value as the character candidate in the sub-path.

III. Character Classification

After character candidates are extracted, they are represented using the MLBP feature [19]. Then the character candidates are verified using an AdaBoost trained classifier [21] that is trained by the following steps. (1) We prepared 56,700 positive and 60,000 negative training images from the ICDAR 2011 training data set, and transformed them to gray-scale images with a size of 18×18. (2) We computed the classification errors for all features and selected the feature with the fewest classification errors as the weak classifier in the current iteration. (3) We updated the weights of the positive and negative samples such that incorrectly-classified samples have higher weights than do correctly classified ones. (4) We used a validation set to check the stop condition. If the iteration satisfied the stop condition we terminated the training; otherwise we updated the weights and returned to step (2). We used two cascades of the classifiers to speed up the classification [22], where each cascade consists of 32 and 64 weak classifiers, respectively. Fig. 3 shows some classification results where the top and bottom row represent the character candidates before and after character classification, respectively. From Fig. 3, it can be said that the character classification process greatly reduced the number of character candidates.

IV. Character Refinement and Text Region Grouping

A. Character Refinement

Character classification still produces incorrect classification results such as false positives or multiple candidates for a single character. To eliminate as many of these incorrect classification results as possible, we propose to use some heuristic rules as follows.

Case 1: We eliminate character candidates that are larger than 80% of the input image size.

Fig. 4. Examples of character candidates that contains internal character candidates.

Case 2: If two character candidates have a similarity value (Eq. (4)) that is greater than 0.5, i.e., their overlapping region is large, we select the candidate with the lowest regularized stability value (Eq. (5)).
Case 3: If one character candidate contains another character candidate internally and the color histogram of the internal character candidate is highly peaked (more than 85% of the total number of pixels) at a specific color, we eliminate the internal character candidate.

Fig. 4 illustrates three examples of *Case 3:* of the character refinement, where the internal character candidates should be eliminated.

B. Text Region Grouping

On some commonly used databases such as the ICDAR competition dataset, character regions must be grouped for evaluation. We used heuristic rules to group neighboring regions together. Two regions R_1 and R_2 are grouped together if they satisfy all the following properties:

1) R_1 and R_2 are similar in width and height.

$$\begin{aligned} width(R_1) &> 0.8 \times width(R_2), \\ width(R_1) &< 1.2 \times width(R_2). \end{aligned} \tag{6}$$

2) The horizontal distance (h_d) between the center points of R_1 and R_2 is within 3 times of the average width of R_1 and R_2.

$$h_d \leq 3 \times \frac{width(R_1) + width(R_2)}{2}. \tag{7}$$

3) The vertical distance (v_d) between the center points of R_1 and R_2 is within a 1/3 of the average height of R_1 and R_2.

$$v_d \leq \frac{1}{3} \times \frac{height(R_1) + height(R_2)}{2}. \tag{8}$$

4) R_1 and R_2 have similar color histogram distributions.

This grouping stage of the proposed text detection method is shown in Fig. 5.

V. Experimental Results and Discussion

We conducted three experiments to compare the recall, precision, and f-score of the proposed text detection method to those of state-of-the-art methods, by performing three different experimentations: The experiments were (1) character-level character candidate extraction, (2) text-level text detection, and (3) step-wise performance of text detection. First, we performed the proposed detection method on the gray channel of the input image; these results are denoted as 'Proposed Method (Gray)'. Then we applied the proposed text detection method to the C_r and C_b channels to improve the detection

Fig. 5. Results of the text region grouping.

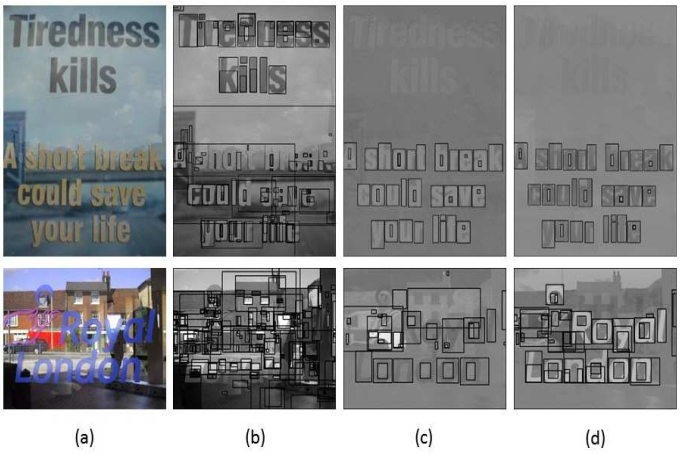

Fig. 6. Two example of the character candidate extraction results of different channels. (a) input image, (b) gray channel, (c) C_r channel, and (d) C_b channel, respectively.

performance further, and combined the text detection results of the gray, C_r and C_b channels; these results are denoted as 'Proposed Method (Gray+C_r+C_b)'. Use of different color channels resulted in extraction of different subsets of characters (Fig. 6).

We used the ICDAR 2011 dataset [23] and the ICDAR 2013 dataset [24] that were used in the Robust Reading Competition (Challenge 2: Reading Text in Scene Images), to evaluate the proposed character candidate extraction method and the proposed text detection method, respectively.

First, we evaluated the character-level recall rates [25] of the proposed candidate extraction method and compared it with other character candidate extraction methods (see Table 1). The character-level recall rate is defined as the proportion of the detected characters among the ground-truth characters, where a character is considered as detected if the area of the bounding box covering the detected character region is matched by more than 90% of the ground-truth region. The proposed character candidate extraction method using three different channels achieved the highest recall rate.

The evaluation method mentioned above requires that over 90% of the detected area is matched by a ground-truth area

TABLE I. CHARACTER-LEVEL RECALL RATES ON THE ICDAR 2011 TEST DATASET.

Algorithm	Character-level Recall rate (%)
Proposed Method (Gray)	99.1
Proposed Method (Gray+C_r+C_b)	99.3
Yin *et al.* [15] (Gray)	90.2
Yin *et al.* (B+G+R)	95.2
Neumann and Matas [25] (Gray)	85.6
Neumann and Mata (Gray+H+S+Gradient)	95.2

TABLE II. CHARACTER-LEVEL RECALL RATES ON THE ICDAR 2011 TEST DATASET (EVALUATION BASED ON SIMILARITY VALUE).

Algorithm	No. of candidates	Character-level Recall rate (%)
ER	1,729,833	89.6
MSER	39,762	53.9
Proposed Method (Gray)	75,124	86.0
Proposed Method (Gray+C_r+C_b)	93,357	87.7

TABLE III. WORD-LEVEL TEXT DETECTION RESULTS ON THE ICDAR 2013 TEST DATA SET.

Algorithm	Recall (%)	Precision (%)	f-score
Proposed Method (Gray+C_r+C_b)	74.23	88.65	80.80
Proposed Method (Gray)	72.01	87.64	79.06
Yin *et al.* [15]	66.45	88.47	75.89
Text_Spotter [11], [25], [26]	64.84	87.51	74.49
CASIA_NLPR [27]	68.24	78.89	73.18
Text_Detector_CASIA [14], [28]	62.85	84.70	72.16

for the detection to be considered correct. Therefore, a small detected region is considered as a correct detection if it is contained within a large ground truth region. To solve this problem, we used our similarity value (Eq. (4)) and regarded the result as a correct detection only if the similarity value between a detected region and ground-truth region was over 0.5. Using this evaluation method, we compared the character candidate extraction method using ERs, MSERs, and our character extraction method (see Table II). The proposed character candidate extraction method using three different channels achieved the highest recall rate.

Second, we compared the recall, precision, and f-scores of the proposed candidate extraction method and other text detection methods (see Table III) using the evaluation method used in the ICDAR 2013 competition [24]. The proposed text detection method using three different channels had higher recall, precision, and f-scores than all existing text detection methods.

Third, we evaluated the text-level recall, precision, and f-score of step-wise detection by the proposed text detection method [24]. As steps were added, recall decreased slowly, but precision and f-score increased rapidly (see Table. IV).

VI. CONCLUSION

This paper presented a new ER-based character candidate extraction method that solves some limitations of using MSERs. The proposed character candidate extraction method divides a local path in an ER-tree into several sub-paths and extracts one character candidate from each sub-path; this process improved the recall rate greatly. We used an AdaBoost trained classifier to verify the extracted character candidates,

TABLE IV. STEP-WISE DETECTION PERFORMANCES OF THE PROPOSED TEXT DETECTION METHOD ON THE ICDAR 2013 TEST DATASET.

Stage	Recall rate (%)	Precision rate (%)	f-score (%)
ER (Gray+C_r+C_b)	85.13	4.89	9.24
ER (Gray)	84.61	5.08	9.58
Character candidate extraction (Gray+C_r+C_b)	78.16	23.18	35.76
Character candidate extraction (Gray)	75.54	23.11	35.39
Character Classification (Gray+C_r+C_b)	74.29	86.41	79.89
Character Classification (Gray)	71.92	86.26	78.44
Character Refinement and Text Grouping (Gray+C_r+C_b)	74.23	88.65	80.80
Character Refinement and Text Grouping (Gray)	72.01	87.64	79.06

then used heuristics based on geometric and color histogram to refine the classified character candidates; these steps improved the precision slightly.

In experiments, the proposed text detection method achieved higher recall, precision, and f-score than all existing detection methods.

In future, we plan to reduce the sensitivity of text detection to blur, which is a major weakness of the detection method that uses connected component analysis. We will also seek a new method that identify characters despite orientation variations in text lines, such as wiggly text line or text that is written in a circle or half-circle.

ACKNOWLEDGMENT

This work was supported by the IT R&D program of MKE/KEIT[10040246, Development of Robot Vision SoC/Module for acquiring 3D depth information and recognizing objects/faces] and also Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2010-0019523).

REFERENCES

[1] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," *Image Processing, IEEE Transactions on*, vol. 9, no. 1, pp. 147–156, 2000.

[2] J. J. Weinman, E. Learned-Miller, and A. R. Hanson, "Scene text recognition using similarity and a lexicon with sparse belief propagation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 10, pp. 1733–1746, 2009.

[3] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 4, pp. 385–392, 2000.

[4] X. 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*, vol. 2. IEEE, 2004, pp. II–366.

[5] 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," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 12, pp. 1631–1639, 2003.

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

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

[8] C. Yi and Y. Tian, "Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification," *Image Processing, IEEE Transactions on*, vol. 21, no. 9, pp. 4256–4268, 2012.

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

[10] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *Image Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 800–813, 2011.

[11] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Computer Vision–ACCV 2010*. Springer, 2011, pp. 770–783.

[12] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[13] 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 *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 2609–2612.

[14] 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*, vol. 34, no. 2, pp. 107–116, 2013.

[15] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 5, pp. 970–983, May 2014.

[16] X. Yin, X.-C. Yin, H.-W. Hao, and K. Iqbal, "Effective text localization in natural scene images with mser, geometry-based grouping and adaboost," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, Nov 2012, pp. 725–728.

[17] L. Gomez and D. Karatzas, "Multi-script text extraction from natural scenes," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, Aug 2013, pp. 467–471.

[18] H. I. Koo and D. H. Kim, "Scene text detection via connected component clustering and nontext filtering," *Image Processing, IEEE Transactions on*, vol. 22, no. 6, pp. 2296–2305, June 2013.

[19] G. Bai, Y. Zhu, and Z. Ding, "A hierarchical face recognition method based on local binary pattern," *Proc. Congr. Image Signal Process*, pp. 610–614, 2008.

[20] M. Donoser and H. Bischof, "Efficient maximally stable extremal region (mser) tracking," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 553–560.

[21] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[22] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[23] A. Shahab, F. Shafait, and A. Dengel, "Icdar 2011 robust reading competition challenge 2: Reading text in scene images," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 1491–1496.

[24] 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 *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1484–1493.

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

[26] ——, "On combining multiple segmentations in scene text recognition," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 523–527.

[27] B. Bai, F. Yin, and C. L. Liu, "Scene text localization using gradient local correlation," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1380–1384.

[28] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2961–2968.