Extraction of Character Regions through Machine Learning and Filtering

Seok-Woo Jang, Sang-Hong Lee

Abstract: ***Characters in images are able to provide main information of the image. Therefore, it is important to analyze various kinds of image data and accurately extract the characters in images. This study proposes a new method of excluding background regions and accurately detecting character regions from input images with the uses of MCT features and Adaboost algorithm. The proposed method first extracts candidate character regions from input images with the uses of MCT features and Adaboost algorithm. It then excludes non-character regions and detects real character regions from the extracted candidate regions with the use of geometrical features. In the experiment of this study, the proposed method more robustly detected character regions from various input color images than a conventional method. For performance comparison, this study compared the method based on existing texture analysis and the proposed method. In this study, to qualitatively evaluate the performance of the proposed method of extracting license plate regions, the accuracy measure was defined. The measure is used to show the ratio of the accurately extracted character regions to all character regions of an image. The conventional method using the frequency factor-based texture information had many errors of character region detection, since it failed to execute binarization of background and character regions properly. On contrary, the proposed method made use of MCT features and Adaboost algorithm, effectively filtered candidate regions with the use of geometrical features, so that it detected character regions more accurately. The proposed character detection method is expected to be usefully applied to the fields of pattern recognition and image processing, such as store sign recognition and license plate recognition.***

Keywords: Filtering, Machine learning, Character data, Feature acquisition, Candidate region.

I. INTRODUCTION

With the remarkable development of hardware and software, the volume of various kinds of image data collected from smartphones, CCTVs, black boxes, drones, satellites, and digital cameras are on the exponential increase. Generally, image big data is recognized as the new next-generation technology capable of analyzing unstructured data beyond conventional structured data, and of extracting meaningful information, and thereby of extracting a new value [1-5].

It is very important to analyze various kinds of image data and extract character regions only from the images [2]. That is because the characters of an image are able to provide the main information that can represent the meaning of the image. In the character extraction field, the image analysis-based extraction of license plates is of special importance. Such extraction is required to make access control in parking lots and speed limit enforcement on roads [3].

As shown in references, there are relevant works of automatic extraction of character regions from various kinds of input images. The method proposed in the study [4] is to split an input image in the N×N square block unit, and then to extract character regions with the use of the point that a character region has a relatively large high-frequency factor in the horizontal or vertical direction. The method proposed in the study [5] is to use saturation data, to create the transition map of the color between the background and character regions, and thereby to extract a character region from an input image. The method proposed in [6] is to detect and track an object with the use of the motions in the gray image shot by a fixed camera, and thereby to detect the license plate of a four-wheeled or two-wheeled vehicle ahead. The method proposed in the study [7] is to detect the license plate region of a vehicle with the use of yellow or green color information in an image. However, this method assumes that the background should be a relatively simple image. Other methods of character region extraction than the aforementioned ones continue to be introduced as references.

The character region detection methods described earlier are able to detect character regions accurately in a certain level. Nevertheless, they have the restriction on the point that they guarantee accuracy only in given surroundings.

Therefore, this study proposes a new algorithm of robustly detecting character regions only from an input image with the uses of MCT features and Adaboost (adaptive boosting) algorithm. Fig. 1 illustrates the overview of the proposed character region detection method.

As shown in Fig. 1, the proposed method is used to extract candidate character regions from an input image with the uses of MCT features and Adaboost learning algorithm, and then filter the extracted candidates with the use of geometrical information and extract real character regions robustly by excluding non-character regions.

This study is comprised as follows. In Chapter 1, the overview of this study is explained. In Chapter 2, how to extract candidate character regions is described. In Chapter 3, through verification, how to select real character regions from candidate character regions is described. In Chapter 4, the result of the experiment for comparing the performance of the proposed method is drawn.

Revised Manuscript Received on July 22, 2019.

***** Correspondence Author

Seok-Woo Jang, Department of Software, Anyang University, Anyang, South Korea. Email: swjang7285@gmail.com

Sang-Hong Lee*, Department of Computer Engineering, Anyang University, Anyang, South Korea. Email: shleedosa@anyang.ac.kr

In Chapter 5, the conclusion of this study and its future research direction are described.

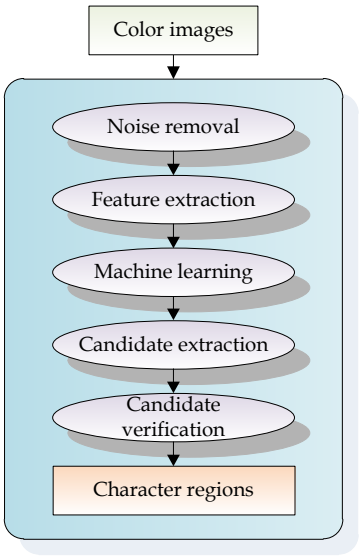


Fig 1. Overall diagram of the proposed approach

II. EXTRACTION OF CANDIDATE AREAS

In this study, MCT (Modified Census Transform) features [8-10] and Adaboost learning algorithm [11-14] are used to extract candidate character regions from an image.

A MCT feature means a local structure feature, expressing the 0 to 1 binary information. In other words, a MCT feature presents the relations between pixel values of a particular position and its adjacent pixel values. Generally, a MCT feature presents the correlations between the pixels adjacent to a particular position, rather than the pixels of the position, in a certain pattern so that it is robust to a change in lighting. In addition, due to its relatively simple calculation, a detection rate is relatively high and an execution time is short in the fields of image processing application like face detection.

In this study, the 3×3 kernel-based MCT feature can be denoted as shown in (1).

$$\Gamma(x) = \otimes_{y \in n'} \xi(\overline{I(x)}, I(y)) \qquad (1)$$

In (1), I(x) represents the pixel value of x, and $\overline{I(x)}$ means the average pixel value of the pixels located in kernel. n' represents a set of pixels adjacent to the center of kernel. $\xi()$, which is comparison function, displays '1' if I(y) is larger than the average pixel value and, otherwise, displays '0'. ⊗, which is decimal conversion operator, converts the 9-digit binary array made as the result of the comparison function $\xi()$ into a decimal number. Therefore, the MCT feature used in this study has a range from 0 to 511. Figure 2 illustrates an example of MCT conversion.

The MCT feature extracted in the previous step is applied to the character detection learner created by Adaboost learning algorithm in order to detect candidate character regions. Adaboost learning as a sort of boosting means the algorithm of making one strong learner with good classification in combination of multiple weak learners with poor classification. Usually, a weak learner means a learner that has 50% of classification accuracy, and a strong learner is referred to as one that has a very small classification error.

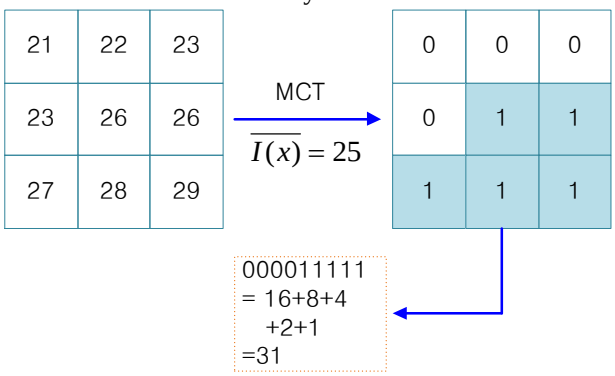


Fig 2. Overall diagram of the proposed approach

Generally, Adaboost algorithm is easy to implement and is fast so that it is widely used in the fields of image processing and pattern recognition [15]. In addition, the algorithm is capable of improving its performance by combining with other kinds of learning algorithms.

Table-I: Pseudo-codes of Adaboost algorithm

Input: learning set $X=\{(x_1,t_1), (x_2,t_2), ..., (x_N,t_N)\}$
Output: classifier ensemble $C=\{(c_k,\alpha_k), 1<k<K\}$

```
C=∅;
for (j=1 to N) w_j=1/N;
for (k=1 to K) {
    Learn c_k considering w_1, w_2, ..., w_N
    ε=0;
    for (j=1 to N)
        if (c_k(x_j)≠t_j) ε=ε+w_j;
    if (ε<0.5)
        α_k=1/2×log((1-ε)/ε)
        for (j=1 to N)
            if (c_k(x_j)≠t_j) w_j=w_j×e^α
            else w_j=w_j×e^-α
        Normalize w_j so that sum of w_1, w_2, ..., w_N should be 1
        C=C∪(c_k,a_k);
}
else {
    c_k=Nil;
    C=C∪(c_k,0);
}
```

Adaboost learning algorithm generates a set from original data according to the error rate of the learned hypothesis, and then learns a new hypothesis and makes a data set on the basis of the hypotheses. It concentrates on hard-to-classify data by reducing a weight value of data with correct prediction and increasing a weight value of data with incorrect prediction according to hypothesis learning and hypothesis error. Actually, Adaboost algorithm has a lower learning error than its theoretical error upper limit. In the algorithm, along with a learning error, a generalization error also decreases.

Table-I shows the pseudo-codes of Adaboost algorithm.

In Table-I, N means the number of samples, w_j is the weight value of the j-th sample, and K is the number of learners. c_k is the k-th learner, ε is an error, and α_k is the reliability of the learner c_k. Adaboost algorithm has the complementary feature of learners. In other words, the weakness of c_k is made up for by c_{k+1}.

III. DETECTION OF REAL CHARACTER REGIONS

The proposed method in this study detects final character regions by verifying the candidate character regions extracted from the Adaboost learning algorithm in the previous step with the use of geometrical features.

The squares that present the character candidates extracted by Adaboost algorithm are intensively found near real character regions. Therefore, first, grouping is performed according to the position and size of squares, and then in each group the squares that are most similar to the average square in terms of position and size are judged to be candidate character regions.

After that, in consideration of the case that detected candidate character regions fails to include all of real character regions, the width and length of a candidate regions are extended as many as k. In this study, k was set to a half (1/2) of the height of candidate character regions.

The adaptive algorithm proposed by Otsu is applied to the extended candidate character regions in order for binarization [23-25]. At this time, a threshold value for binarization should be a value higher than the average pixel value of a candidate character, because the background in a candidate character region (e.g., a license plate region) generally has a brighter color than a character.

Labeling [26-28] is applied to the candidate regions of binary-coded characters. If the size of a labeled region is too small or does not fit the size of a license plate, the region is removed from candidate character regions. After that, corner points are detected from a labeled region. Of the extracted corner points, four angular points located at the outer most are selected.

To extract a corner point [29-31], this study made use of FAST which is the fast corner extraction algorithm proposed by Edward Rosten at University of Cambridge, the UK [32-35]. As guessed in its name, FAST is the method of extracting feature points with fast speed. The good thing of FAST is that FAST has far better quality than other methods, even if it has the optimized design for speed. The basic process of FAST algorithm is presented in Fig. 3.

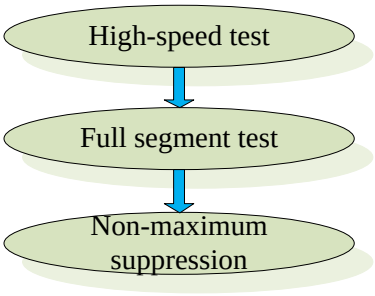


Fig 3. Basic process of FAST algorithm

FAST algorithm has three stages as shown in Fig. 3. In the first stage, corner candidates are determined fast. In this stage, the distance from the point P and its neighboring points is checked. In the second stage, the brightness of each one of corner candidates is analyzed. The brightness is judged on the basis of the values of 16 pixels in the circle that has the radius 3 from the corner candidate point P. In short, the brightness difference between P and its neighboring points is analyzed; if there are more than k points which are darker or brighter, the point P is judged to be a corner point.

In the third stage, 3×3 mask is applied at the point P, and the point with the largest difference in the range is selected as the final corner point. The problem that FAST corner has is that if a certain point P is recognized as a corner point, the points adjacent to the point P are often detected as corner points. Accordingly, to solve the problem, FAST applies the third stage as post-processing.

As for the candidate character region that has an angular point detected, the length of the left side are compared with that of the right side in a license plate, as shown in (2), and the car license plate is finally verified.

$$\begin{aligned} &IF\ (|Len_l - Len_r| < (Len_l + Len_r)\cdot 0.2)\ THEN \\ &\quad R_i\ is\ a\ character\ region \\ &ELSE \\ &\quad R_i\ is\ a\ non-character\ region \end{aligned} \tag{2}$$

In (2), Len_l represents the length of the left side of a candidate square, and Len_r means the length of the right side. R_i means the i-th candidate character region.

IV. EXPERIMENTAL RESULTS

The computer used in the experiment of this study has Intel Core(TM) i7 2.93Ghz CPU, 8GB main memory, and Windows 7 operating system. Microsoft Visual Studio and OpenCV (open library) were used to implement the proposed algorithm. To evaluate the performance of the proposed algorithm, many different kinds of test images including character regions were collected and used. In this study, learning is processed with the uses of 307 license plate images and 2,736 background images.

In this study, to qualitatively evaluate the performance of the proposed method of extracting license plate regions, the accuracy measure of (3) was utilized. The measure is used to show the ratio (%) of the accurately extracted character regions to all character regions of an image. In (3), $NO_{detected}$ is the number of the character regions detected accurately with the use of the proposed algorithm, and $NO_{existing}$ is the number of all character regions in a test image.

$$Accuracy = \frac{NO_{detected}}{NO_{existing}} \times 100(\%) \tag{3}$$

Fig. 4 displays the graphs of performance comparison evaluation in terms of accuracy of the character region extraction algorithm based on (3). As shown in Fig. 4, the proposed algorithm detected character regions more accurately. The conventional method using the frequency factor-based texture information had many errors of character region detection, since it failed to execute binarization of background and character regions properly. On contrary, the proposed method made use of MCT features and Adaboost algorithm, effectively filtered candidate regions with the use of geometrical features, so that it detected character regions more accurately.

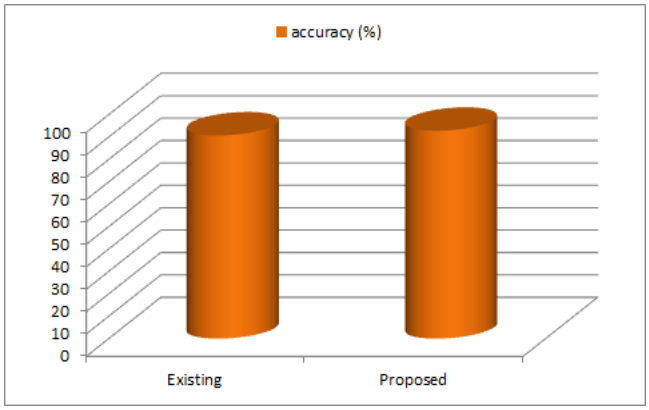


Fig 4. Performance comparison

V. CONCLUSION

These days, more demands are made for the research to analyze indoor and outdoor images and then robustly segmenting the character regions in images like license plate images.

This study proposed the new method of excluding background regions and robustly detecting character regions from various input images, with the uses of MCT features and Adaboost algorithm. In the proposed method, candidate character regions were first extracted with the uses of MCT features and Adaboost learning algorithm. The extracted candidate character regions were then filtered with the use of geometrical features so as to exclude non-character regions and robustly detect real character regions. In the experiment of this study, the proposed method detected character regions more accurately than the conventional method.

The future research plan is to apply this proposed character region detection method to more various types of indoor and outdoor input images in order to evaluate the proposed algorithm in diverse circumstances, and to add a character recognition module to the currently developed character region detection algorithm.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT) (2019R1F1A1056475).

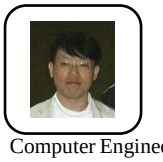
REFERENCES

1. A M. Aladwani, “Compatible quality of social media content: conceptualization, measurement, and affordances,” Int. J. Inform. Manage., 37(6), 2017, pp. 576-582.
2. S. Wazarkar, B. N. Keshavamurthy, “A survey on image data analysis through clustering techniques for real world applications,” J. Vis. Comm. Image Represent., vol. 55, 2018, pp. 596-626.
3. C. Qin, X. Qian, W. Hong, X. Zhang, “An efficient coding scheme for reversible data hiding in encrypted image with redundancy transfer,” Inform. Sci., vol. 487, 2019, pp. 176-192.
4. Q. Zhang, Y. Shang, Y. Wang, Y. Liu, N. Wang, Z. Gui, G. Yang, “Denoising for low-dose CT image by discriminative weighted nuclear norm minimization,” IEEE Access, vol. 6, 2018, pp. 46179-46193.
5. A. V. Opbroek, M. A. Ikram, M. W. Vernooij, M. D. Bruijne, “Transfer learning improves supervised image segmentation across imaging protocols,” IEEE Trans. Med. Imag., vol. 34, no. 5, 2015, pp. 1018-1030,
6. K. Fu, J. Li, J. Jin, C. Zhang, “Image-text surgery: efficient concept learning in image captioning by generating pseudopairs,” IEEE T. Neur. Net. Lear., vol. 29, no. 12, 2018, pp. 5910-5921.
7. D. NguyenVan, S. Lu, S. Tian, N. Ouarti, M. Mokhtari, “A pooling based scene text proposal technique for scene text reading in the wild,” Pattern Recogn., vol. 87, 2019, pp. 118-129.
8. G. J. Ansari, J. H. Shah, M. Yasmin, M. Sharif, S. L. Fernandes, “A novel machine learning approach for scene text extraction,” Future Gener. Comp. Sy., vol. 87, 2018, pp. 328-340.
9. A. Onan, S. Korukoglu, H. Bulut, “Ensemble of keyword extraction methods and classifiers in text classification,” Expert Syst. Appl., vol. 57, 2016, pp. 232-247.
10. M. R. Asif, Q. Chun, S. Hussain, M. S. Fareed, S. Khan, “Multinational vehicle license plate detection in complex backgrounds,” J. Vis. Comm. Image Represent., vol. 46, 2017, pp. 176-186.
11. X. Qian, G. Liu, H. Wang, R. Su, “Text detection, localization, and tracking in compressed video,” Signal Process. Image Comm., vol. 22, no. 9, 2007, pp. 752-768.
12. W. Kim, C. Kim, “A new approach for overlay text detection and extraction from complex video scene,” IEEE T. Image Process., vol. 18, no. 2, 2009, pp. 401-411.
13. H. Lee, S. Chen, S. Wang, “Extraction and recognition of license plates of motorcycles and vehicles,” in 17th IEEE International Conference on Pattern Recognition. Cambridge, UK, 2004, pp. 356-359.
14. K. Deb, K. Jo, “HSI color-based vehicle license plate detection,” in International Conference on Control, Automation and Systems, Seoul, South Korea, 2008, pp. 687-691.
15. L. Bai, J. Liang, C. Dang, F. Cao, “A novel attribute weighting algorithm for clustering high-dimensional categorical data,” Pattern Recogn., vol. 44, no. 12, 2011, pp. 2843-2861.
16. Y. G. Lee, Z. Tang, J. N. Hwang, “Online-learning-based human tracking across non-overlapping cameras,” IEEE T. Circ. Syst. Vid., vol. 28, no. 10, 2018, pp. 2870-2883.
17. X. Chang, L. Jiao, F. Liu, F. Xin, “Multicontourlet-based adaptive fusion of infrared and visible remote sensing images,” IEEE Geosci. Remote S., vol. 7, no. 3, 2010, pp. 549-553.
18. B. Sun, S. Chen, J. Wang, H. Chen, “A robust multi-class AdaBoost algorithm for mislabeled noisy data,” Know.-Based Syst., vol. 102, 2016, pp. 87-102.
19. C. Gao, P. Li, Y. Zhang, J. Liu, L. Wang, “People counting based on head detection combining Adaboost and CNN in crowded surveillance environment,” Neurocomputing, vol. 208, 2016, pp. 108-116.
20. H. Dogan, O. Akay, “Using AdaBoost classifiers in a hierarchical framework for classifying surface images of marble slabs,” Expert Syst. Appl., vol. 37, no. 12, 2017, pp. 8814-8821.
21. T. Wang, Z. Qin, Z. Jin, S. Zhang, “Handling over-fitting in test cost-sensitive decision tree learning by feature selection, smoothing and pruning,” J. Syst. Software, vol. 83, no. 7, 2010, pp. 1137-1147.
22. A. Fernandes, A. Utkin, J. Eiras-Dias, J. Silvestre, P. Melo-Pinto, “Assessment of grapevine variety discrimination using stem hyperspectral data and AdaBoost of random weight neural networks,” Appl. Soft Comput., vol. 72, 2018, pp. 140-155.
23. H. Cai, Z. Yang, X. Cao, W. Xia, X. Xu, “New iterative triclass thresholding technique in image segmentation,” IEEE T. Image Process., vol. 23, no. 3, 2014, pp. 1038-1046.

24. X. C. Yuan, L. S. Wu, Q. Peng, "An improved Otsu method using the weighted object variance for defect detection," Appl. Surf. Sci., vol. 349, 2015, pp. 472-484.
25. A. M. A. Talab, Z. Huang, F. Xi, L. HaiMing, "Detection crack in image using Otsu method and multiple filtering in image processing techniques," Optik, vol. 127, no. 3, 2016, pp. 1030-1033.
26. L. He, X. Zhao, Y. Chao, K. Suzuki, "Configuration-transition-based connected-component labeling," IEEE T. Image Process., vol. 23, no. 2, 2014, pp. 943-951.
27. L. Ma, X. P. Zhang, J. Si, G. P. Abousleman, "Bidirectional labeling and registration scheme for gray scale image segmentation," IEEE T. Image Process., vol. 14, no. 12, 2005, pp. 2073-2081.
28. L. He, Y. Chao, K. Suzuki, "Two efficient label-equivalence-based connected-component labeling algorithms for 3-D binary images," IEEE T. Image Process., vol. 20, no. 8, 2011, pp. 2122-2134.
29. G. V. Pedrosa, C. A. Z. Barcelos, "Anisotropic diffusion for effective shape corner point detection," Pattern Recogn. Lett., vol. 31, no. 12, 2010, pp. 1658-1664.
30. P. Sathiya, P. Anandhakumar, "Probabilistic collision estimation for tracked vehicles based on corner point self-activation approach," Comput. Electr. Eng., vol. 74, 2019, pp. 557-568.
31. O. Haggui, C. Tadonki, L. Lacassagne, F. Sayadi, B. Ouni, "Harris corner detection on a NUMA manycore," Future Gener. Comp. Sy., vol. 88, 2018, pp. 442-452.
32. S. K. Lam, G. Jiang, M. Wu, B. Cao, "Area-time efficient streaming architecture for FAST and BRIEF detector," IEEE T. Circuits-II: Express Briefs, vol. 66, no. 2, 2019, pp. 282-286.
33. Y. Xing, D. Zhang, J. Zhao, M. Sun, W. Jia, "Robust fast corner detector based on filled circle and outer ring mask," IET Image Processing, vol. 10, no. 4, 2016, pp. 314-324.
34. M. Awrangjeb, G. Lu, C. S. Fraser, "Performance comparisons of contour-based corner detectors," IEEE T. Image Process., vol. 21, no. 9, 2012, pp. 4167-4179.
35. S. K. Lam, T. C. Lim, M. Wu, B. Cao, B. A. Jasani, "Area-time efficient FAST corner detector using data-path transposition," IEEE T. Circuits-II: Express Briefs, vol. 65, no. 9, 2018, pp. 1224-1228.

AUTHORS PROFILE

Seok-Woo Jang received the B.S., M.S., Ph.D. degrees in Computer Science from Soongsil University, Seoul, Korea, in 1995, 1997, and 2000, respectively. From October 2003 to January 2009, he was a Senior Researcher with the Construction Information Research Department at Korea Institute of Construction Technology (KICT), Ilsan, Korea. Since March 2009, he has been a Professor in the Department of Software, Anyang University, Korea. His primary research interests include robot vision, augmented reality, video indexing and retrieval, cluster computing, biometrics and pattern recognition.



Sang-Hong Lee received the B.S., M.S., and Ph.D. degrees in Computer Science from Gachon University, Korea in 1999, 2001, and 2012, respectively. He is currently an assistant professor in the Department of Computer Engineering at Anyang University, Korea. His research focuses on neuro-fuzzy systems, stocks prediction systems, and biomedical prediction systems.