International Journal of Engineering & Technology, 7 (2.12) (2018) 29-34

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Study on the extraction of the text region from natural scene images by an analysis of the edge-oriented pattern

Jae-Ho Yang [1] *, Gang-Seong Lee [2], Young-Pyo Hong [3], Sang-Hun Lee [2]

[1] *Department of Plasmadiodisplay, Kwangwoon University, Seoul 01897, Korea*
[2] *Ingenium College of Liberal arts, Kwangwoon University, Seoul 01897, Korea*
[3] *Department of Hospital Management, International University, Jinju 52833, Korea*
**Corresponding author E-mail: 2016143701@kw.ac.kr*

Abstract

Background/Objectives: In this paper, we propose a hybrid scene-detection method using an edge and textural analysis in natural scene images, and finally, we detect the text regions by removing the non-text regions through a pattern analysis of each region.
Methods/Statistical analysis: The proposed algorithm is divided into the pre-processing stage and the extraction processing stage to perform the text detection. The lost texts that are improved through a histogram equalization for the minimization of the loss of the text parts that is due to light exposure are detected before the edge detection. After that, the edge is detected using the Canny operator. The detected edge is obtained in the step of applying the SWT algorithm to detect the text candidate regions. The extraction processing step is the step of removing the noise region that is detected by the pixel analysis of the SWT algorithm, and it analyzes the pattern of the text regions and then removes the non-text regions to finally detect the text regions. For the quantitative comparison of the proposed algorithm, our results are compared with the ground-truth image using the precision, recall, and F-measure.
Findings: One of the existing text-detection algorithms, the edge-based method, is problematic, as, in addition to the text, the complex backgrounds and textures are detected as the edges in natural scene images. The connected component-based method is also problematic, as the non-text region is included in the text region in the process of finding the connection component.
Improvements/Applications: The proposed method shows an effective text-detection result regardless of the light exposure in natural scene images compared with the conventional text-detection algorithm.

Keywords: *Text Detection; Histogram Equalization; SWT Algorithm; Edge-Based Method; Texture-Based Method; Pattern Analysis.*

1. Introduction

The texts of documents and images are important information for an image analysis, and a text algorithm is required for the text detection[1]. Accordingly, various text-recognition studies are being conducted. Document images can be divided into artificial text[2] and scene text[3]. In the case of artificial text, the texts that are embedded randomly in multimedia such as films, news media, movie subtitles, and movie posters are located in a fixed position, or the variables are small due to the external factors. However, the scene text that is an image (traffic sign, signboard, etc.) that is captured by a mobile phone in daily life is detected by various external factors such as the text direction, size, style, and color position, making it difficult to detect. To solve this problem, numerous methods for the text detection regarding natural scene images have been recently proposed.
The typical methods of text detection are the region-based method, the edge-based method, and the texture-based method. The region-based method divides the input image using information such as the color and the texture, and then the input image is divided into the text regions and the non-text regions using the specific conditions of each connection. The implementation of this method is simple, and it is an efficient and widely used method for noisy and low-resolution images. The edge-based method is generally a method of obtaining the edge of an object by using the brightness differences in the image, and it shows a result difference depending on the type of operator that is used.
The texture-based method obtains the features by using a Gabor filter or a wavelet transform in order to take advantage of the textural property of a text region, and it is a method of detecting a text region by searching for a text pattern or a specific texture. For the proposed hybrid method of this paper, edge-based and texture-based methods are used to find the text candidate regions, and also to remove the non-text regions through an analysis of the patterns in each region. The final result is the detection of the text regions.

2. Text-detection method

The text detection regarding images should take into account the background complexity, as well as the variety of font sizes, styles, orientations, and alignments; as a result, various text-detection approaches have been proposed. This type of approach is used to distinguish between the text and non-text areas using the various attributes that are associated with the texts in the image such as the color, intensity, area, connected components, textures, and edges. This chapter describes the edge-based, texture-based, and connected component-based methods that are the typical types of the text-detection method.

2.1. Edge-based method

Edge-based method[4] is generally used to calculate the edge of an object through the difference of the brightness between the text or the surrounding pixels of the object in the image using the first-derivative operation (Sobel, Prewitt, Robert)[5]or the second-derivative operation (Laplacian, Canny). This method is suitable for finding the internal text symmetry, but it is also disadvantageous, as it detects an erroneous region in a complex background or textural image.

2.2. Texture-based method

Texture-based method[6]is a method of detecting the pattern shape of a text or a specific texture for which the text and background are detected using the textural property, pattern, and texture of the image. Fourier transform or the wavelet transform is used to extract the text. Texture-based methods are not affected by the image resolution, noise, illumination changes, or tilt, but the texture-classification calculation is complicated, and the disadvantage of the inaccuracy of the same-area text detection is problematic.

2.3. Connected component-based method

Connected component-based method[7] is a method of grouping the text lines around a connected area of an image to create a bounding box and to detect the text candidates. This method uses the text geometric information and is not affected by changes such as the text rotation, brightness, text color change, and resizing. This method uses a morphology technique to detect the text regions. However it is difficult to detect the text regions in a low-resolution or complex image.

3. Preprocessing process

A natural scene image is an image that is taken in daily life. To minimize the loss of the text parts that is due to light exposure, which is a problem in the text-extraction process regarding natural scene images, the lost texts are improved by a histogram equalization that is performed prior to the edge detection. The width of the stroke inside the edge is obtained by applying the Stroke Width Transform (SWT) algorithm for the edge detection. After that, Detection of the text candidate regions is achieved by filling the inside of text-region candidate using the gradient value. [Figure 1] shows the preprocessing-process step.

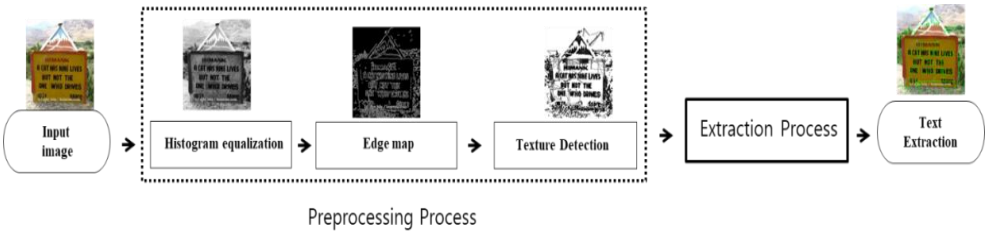

Fig. 1: Preprocessing-Step Process.

3.1. Histogram equalization

Texts are lost from natural scene images due to light exposure. To minimize this loss, histogram equalization[8] is used as the preprocessing in this study. Histogram equalization is an image-enhancement task that produces a uniform distribution of the intensity values of the image pixels. The formulae are as follows:

$$\sum[i] = \sum_{j=0}^{i} hist[j] \quad (1)$$

$$\mathrm{n[i]} = \sum[i] \times \frac{1}{N} \times I_{max} \quad (2)$$

The histogram equalization is performed by obtaining the cumulative-frequency values from 0 to i, as in Equation (1). The cumulative value is normalized using Equation (2), where N is the total number of pixels in the image and the I_{max} is 255. In this process, the pixel value i is transformed into the normalized value n [i] and the resultant image is generated.

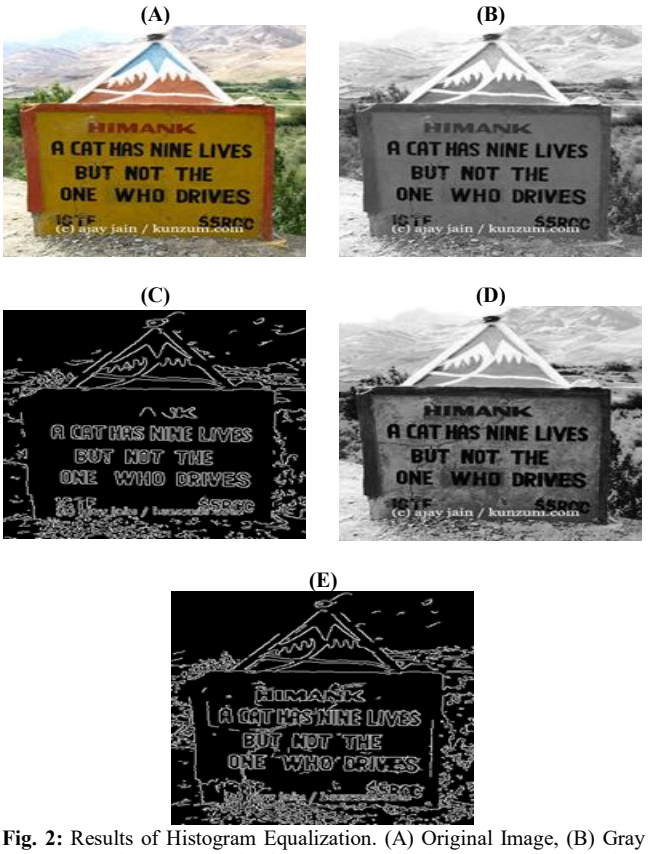

Fig. 2: Results of Histogram Equalization. (A) Original Image, (B) Gray Image, (C) Edge-Map Result of (B), (D) Result of Histogram Equalization, and (E) Edge Map Result of (D).

[Figure 2] is the result of the edge-map generation using the Canny operator after the histogram equalization. The image in (b) is an image that has been subjected to a grayscale conversion for the

extraction of the edge map, (c) is an edge map that has been obtained using the Canny operator, (d) is the result of the performance of the histogram equalization for the minimization of the text loss that is due to light exposure, and (e) is the result of finding the edge map through the application of the Canny operator after the histogram equalization. From the results of Fig. 2, it can be seen that, after the histogram equalization, the result of (e) shows a decrease of the loss of the text data that is greater than the result of (c).

3.2. SWT (stroke width transform)

The SWT algorithm[9,10] is used to calculate the edge width using the texts based on the edge-based image, for which a line is drawn using the value of the gradient (a), and this is followed by the calculation of the internal filling algorithm. The SWT algorithm is used for an analysis of the pattern inside the text of the edge-map result.

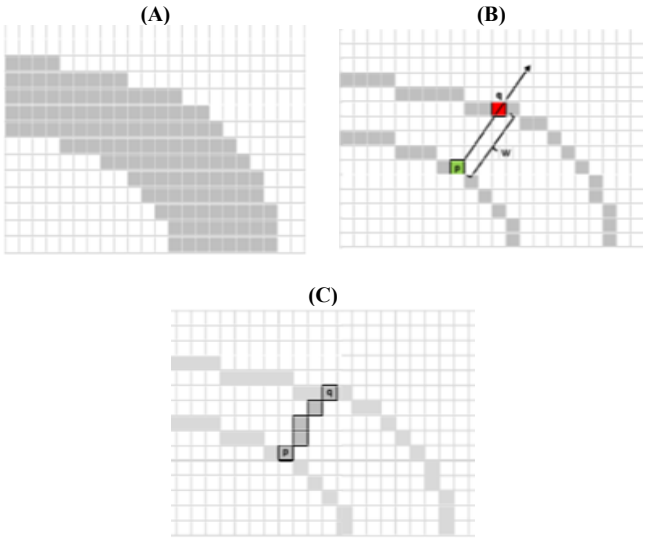

Fig. 3: SWT Algorithm. (A) Gradient, (B) Searching in the Direction of the Gradient(C) Found Stroke Width.

[Figure 3] is a method of constructing the SWT algorithm. Based on the edge map that is obtained by the Canny operator, the gradient direction of the edge pixel p is determined. Then, the width is calculated by finding q, and the inside is filled with a single image if the width is constant. Figure 4 shows the results of the application of the SWT algorithm. As shown in the figure, the width of each edge of the edge map, which is derived using the Canny operator and fills the inside, can be calculated.

Fig. 4: Results of the SWT Algorithm. (A) Original Image, (B) Edge Map, (C)Result of Applying (B) to the SWT Algorithm, (D) Histogram Equalization of the Edge Map,And (E) Result of Applying (D) to the SWT Algorithm.

[Figure 4] shows the results of the application of the SWT algorithm, where the application of the SWT algorithm after the histogram equalization (d) resulted in a lesser text loss compared with (c).

4. Proposed method

For the extraction processing step, the noise region that is erroneously detected through the analysis of the pixels of the SWT result must first be removed. After that, the division of each area is performed through a labeling procedure. Finally, an analysis of the patterns of the text areas such as the height, width, and pixel frequency of each area is performed, followed by the detection of the text areas through the removal of the non-text areas.[Figure 5] shows the Extraction-processing step.

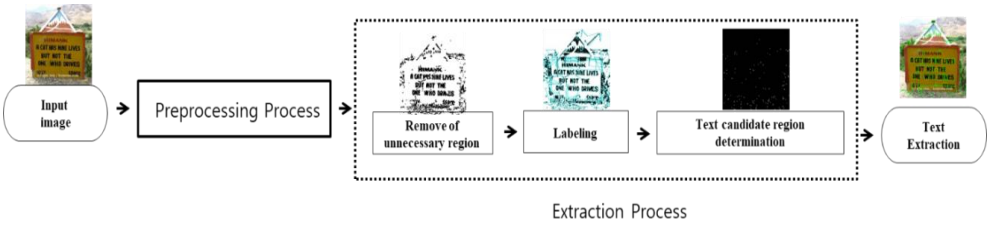

Fig. 5: Extraction Processing Step.

4.1. Noise removal of the SWT results

Before the judging of the non-text and text regions, a binarization was performed to remove the noise from the SWT results. When the SWT was analyzed, the pixel value of the text area is low when the background is brighter than the text area, and a high pixel value is evident when it is a non-text area. After an analysis of the distribution of the values of the entire pixels of the image via [Figure 6], the P_{value} that is divided into the noise was found.

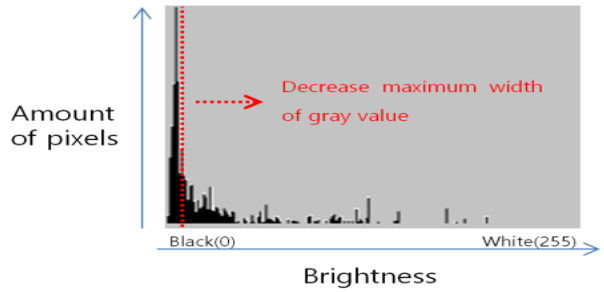

Fig. 6: Analysis of the SWT Histogram for A Binarization.

$$P_{(x,y)} = \begin{cases} P_{(x,y)}, & P_{(x,y)} > P_{value} \\ 0 & else \end{cases} \quad (3)$$

Where $P_{(x,y)}$ is the pixel value of each location and P_{value} is the value with the largest decreasing width of all of the pixel values. [Figure 7] shows the results of the removal of the noise of the SWT algorithm.

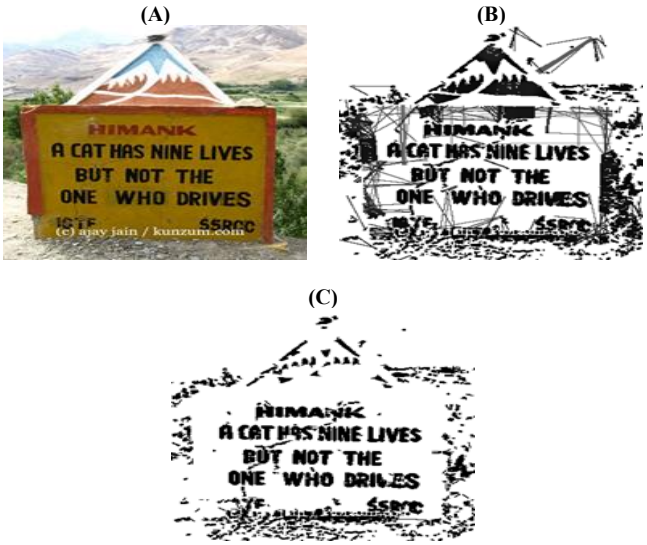

Fig. 7: Results of the Removal of the Noise of the SWT Algorithm. (A) Original Image, (B) SWT Result, and (C) Noise-Removal Result.

4.2. Labeling

In order to analyze the characteristics of the text regions in the SWT-algorithm results with the noise removed, each result is labeled. For the labeling, the rectangle box method is generally used. This method can only label parts of the area, and this creates an excessive margin during the computation of the proportion of the pixels by area under the text-judgment condition. To solve this problem, the shape of each area was labeled. In the case of the rectangle box method, the position of each region is obtained using the contour, and each of the regions is represented by a rectangular shape. Shape labeling is a contour-labeling method.

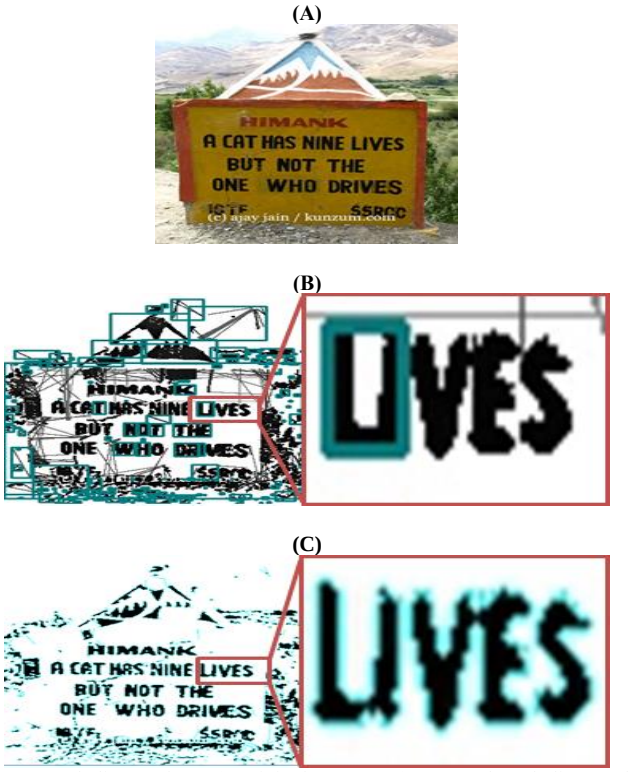

Fig. 8: Labelingresults. (A) Original Image, (B) Rectangle Labeling, and (C) Shape Labeling.

However, a part of the text areas is not designated accordingly in the labeling process of (b), and this problem is shown in [Figure 8]. But in (c), all of the text regions are included in the labeling.

4.3. Determination of the text candidate area

After the labeling process, the patterns of non-text regions were analyzed by inverting the image to remove the non-text regions.

$$\theta = \frac{\sum_{i=1}^{n}\left(\frac{Nop}{Area}\right)}{Noa} \quad (4)$$

$$T_{area} = \begin{cases} Area_{Nop} > \theta \\ else \quad NT \end{cases} \quad (5)$$

In Equation (4), Noa is the total number of regions and Nop is the number of pixels in each region. Using these values, the number of pixels was divided into non-text areas when the number of pixels that are included in each area is smaller than the average value.

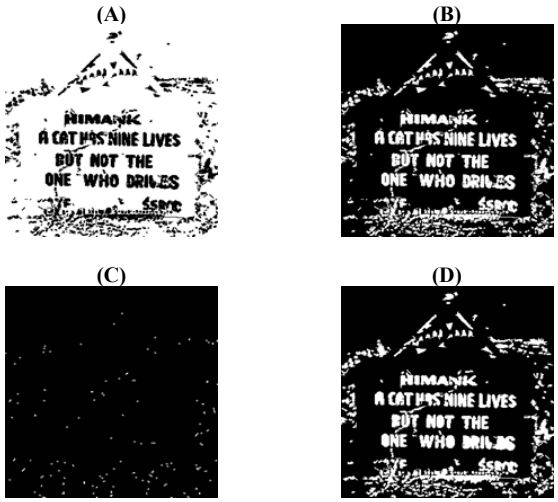

Fig. 9: Results of the Determination of the Text Candidate Region Using the Pixelcount. (A) Modified SWT Result, (B) Inverse of the SWT Image, (C) Non-Text Area (D) Result.

In [Figure 9], when the frequency of the pixels of each region in the noise-data-removal result is small, the results of the removal are shown by the separation of the frequency into the non-text areas. Next, to remove the non-text regions, rectangle labeling was applied to each height and width, and then the following equation was used:

$$T_{area} = \begin{cases} A_h < H_{avg} || A_w < W_{avg} \\ \text{else} \end{cases} \begin{matrix} \\ \text{NT} \end{matrix} \quad (6)$$

Where T_{area} is the text area, $A_h$is the average height of each area, and $A_w$is the average width of each area. Figure 10 shows the results of the removal of the non-text areas that is achieved using Equation (6).

Fig. 10: Result of the Determination of the Text Candidate Region Using the Average Height and Width.

[Figure 10] shows the result of the division of the non-text areas by the calculation of the width and the height of each area. The areas that are smaller than the average of the respective regions as a result of the expressions of (5) and (6) are represented by (b). From the final result of (c), the result that is judged as the non-text region and that does not contain the region is evident.

5. Experiment results

In this paper, the experiments were performed using the MSRA Text Detection 500 (MSRA-TD 500) database, which is a public dataset that is used for text-detection research, and the images were taken using a camera. The experiment was developed using OpenCV as a Visual Studio 2013 program in the Windows 7 environment. The following figures are the results from the MSRA-TD 500 images.

Fig. 11: Results of the Sign Image.

[Figure 11] shows the results of the text-area detection in the images with the small text size. These results show that even if the ratio of the texts that occupy the image is small or of a small size, the colors of the histogram equalization are different from the background; therefore, the text regions are not lost and are finally detected.

Fig. 12: Results of the Sign Image (2).

[Figure 12] shows the experiment results of an image with a larger text size than that of the Fig. 9 image. Here, the results show that the gradient and the edge of the text area are not noticeably lost compared to the background, but the text areas are detectable even in the image with a large text size in the image.

Fig. 13: Result of the Outdoor Image.

The images of [Figure 13] do not show sound results, as an area was wrongly detected and a part of the text was not detected. The results of the proposed method show that the text color is not constant, or a part of the text area is lost in the process of the edge calculation, and this is not included in the final area. In addition, the results of the third row show that the results that are included in the final region contain non-text regions, or the text regions that are smaller than the other text regions in the image were not detected.

Fig. 14: Additional Examples of the Camera Images.

[Figure 14] shows the results of the algorithmic execution according to the camera images containing the texts that can be seen in real life. The SWT algorithm was used with the edge and the gradient to detect all of the text candidate regions, and then the non-text regions were removed using the text-region pattern analysis for the detection of the final text regions in the various texts regardless of whether they are English, Chinese, or Korean.

The experiment results show that the proposed algorithm minimized the loss of light that is due to light exposure in natural scene images with complex backgrounds and various patterns, and it improves the detection of the text regions. To evaluate the performance of the proposed algorithm, the accuracy was calculated according to the precision and the recall while the F-measure is a single measure of the algorithm, and a sound performance was shown using the two measures. In Equations (7) and (8), TP is a true positive, E is the estimated rectangles, and T is the ground-truth rectangles, as follows:

$$\text{Precision} = \frac{|\text{TP}|}{|E|}, \quad (7)$$

$$\text{Recall} = \frac{|TP|}{|T|}, \quad (8)$$

$$f = 2 \times \text{precision} \times \text{recall}/(\text{precision} + \text{recall}). \quad (9)$$

Table 1: Evaluated Performances of Different Text-Detection Methods

Method	Precision	Recall	F-measure
Proposed Method	0.74	0.66	0.69
Kang et al.11	0.71	0.62	0.66
Yao et al. 12	0.63	0.63	0.60
TD-ICDAR	0.53	0.52	0.50
Epshtein et al. 13	0.25	0.25	0.25
Chen et al. 14	0.05	0.05	0.05

[Table 1] shows the average values of the precision, recall, and F-measure of each algorithm using Equations (7), (8), and (9). In this paper, the proposed new algorithm filtered out the non-text regions after the detection of all of the text candidate regions. Therefore, the precision, recall, and F-measure results were improved. However, when the light exposure in the external image is severe or the text size is not constant, the process of judging a small text area as a non-text area in the process of the removal of the area is inaccurate.

6. Conclusion

The proposed method of this study removes the non-text areas through a pattern analysis for the accurate text detection in natural scene images for which a hybrid edge–texture method is used for the text detection. The proposed method minimized the texts loss that is due to light exposure through histogram equalization by implementing a preprocessing for the text extraction in the natural scene images, for which the SWT algorithm was used to fill the interior of the text candidate regions for the pattern analysis of the extraction processing stage. In the extraction process, the height, width, and area of each contour are calculated during the image-pattern analysis, and by applying the formulae, each region is divided into text and non-text regions. The final result shows that the text regions were detected in various languages (English, Korean, Chinese, etc.) regardless of the size of the text. In the SWT processes, however, the text areas were lost, or the non-text areas such as the background area were also detected as a sentence area. In the future, it is expected that a more efficient text extraction will be possible through a supplementation of texts of the same texture as a single area.

7. Acknowledgment

This study was sponsored by the 2017 research fund of Kwangwoon University.

References

[1] C.P.Sumathi, T.Santhanam, N.Priya, “Techniques and challenges of automatic text extraction in complex images: a survey”, Journal of Theoretical and Applied Information Technology, Vol. 35, No. 2, 2012, pp. 225-235.

[2] Raza A, Siddiqi I, Djeddi C, Ennaji A,"Multilingual artificial text detection Using a Cascade of Transforms" In Proceeding of 2013 12th International Conference on Document Analysis and Recognition (ICDAR), 2013, pp. 309–313.

[3] Tahani Khatib, Huda Karajeh, Hiba Mohammad, Lama Rajab,“A hybrid multilevel text extraction algorithm in scene images" Scientific Research and Essays, 2015, pp.105-113.

[4] C. Liu, C. Wang, R. Dai, "Text detection in images based on unsupervised classification of edge-based features", Proc. IEEE Int. Conf. Doc. Anal. Recognition, 2005, pp. 610-614.

[5] G. S. Lee, J. H. Park, J. S. Kim, S. H. Ryu, S. H. Lee, “Independent Object Tracking from Video using the Contour Information in HSV Color Space”, Indian Journal of Science and Technology, 2016, pp.1-8.

[6] Ji R, Xu p, Yao H, Zhang Z, Sun X, Liu T, “Directional Correlation Analysis of Local Haar Binary Pattern for Text Detection.” Proceeding of the International Conference on Multimedia and Expo, 2008, pp.885-888.

[7] H. Koo, D. H. Kim, "Scene text detection via connected component clustering and non-text filtering", IEEE Trans. Image Process, 2013, pp. 2296-2305.

[8] G. S. Lee, J. H. Park, S. H. Lee, “A Study on the Convergence Technique enhanced GrabCut algorithm using color histogram and modified sharpening filter”, Korea convergence society, 2015, pp. 1-8.

[9] Y. Feng, Y, Song, Y, Zhang, “Scene Text Detection Based on Multi-Scale SWT and Edge Filtering”, Pattern Recognition (ICPR), 2016.

[10] B. Epshtein, E. Oyek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In CVPR, 2010, pp.2963-2970.

[11] Le Kang, Yi Li, David Doermann, “Orientation Robust Text Line Detection in Natural Images”, Vision and Pattern Recognition (CVPR), 2014, pp. 1-8.

[12] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu, “Detecting texts of arbitrary orientations in natural images,” Vision and Pattern Recognition(CVPR),2012pp. 1083–1090.

[13] Boris Epshtein, Eyal Ofek, and Yonatan Wexler, “Detecting text in natural scenes with stroke width transform,” Vision and Pattern Recognition (CVPR), 2010, pp. 2963–2970.

[14] Xiangrong Chen and Alan L Yuille, “Detecting and reading text in natural scenes,” Vision and Pattern Recognition (CVPR), 2004pp. 359–366.