International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

Effective Image Quality Inspection Based on Color and Texture Features

Seok-Woo Jang, Sang-Hong Lee

Abstract*: *As people's interest in software increases, the number and types of newly developed applications increase day by day. Therefore, there are attempts to automatically check whether the newly developed interface of these programs works correctly through image processing instead of manually. In this paper, we describe a strategy to robustly detect execution error screens of applications through comparative analysis of key features from various kinds of image scenes. In the approach described in this study, we extract the main multiple features representing the image. Then, by comparing the difference of the extracted multiple features, it is effectively determined whether the image is the same normal image as the reference image or an error image similar to the target image but different from each other. Experimental results verify that the described algorithm accurately detects normal and faulty images by comparing the main multiple features from various types of images. For performance evaluation, we quantitatively compared the performance of the introduced dissimilar region extraction strategy with those of other conventional methods. The existing methods of extracting similarity between images based on one feature or histogram comparison mostly lack information to sufficiently express the entire image. Therefore, when similar areas between two images are extracted, many errors occurred due to incorrect matching. However, the proposed multiple feature-based method obtains richer information by extracting multiple features from the input image. In addition, because the matching is done based on this information, the accuracy is relatively high. The approach introduced in this study is expected to be effectively used in many practical applications related to image processing such as image retrieval, trademark comparison, video security, application quality inspection, etc.

Keywords*: *Faulty region, Image processing, Color model, Texture feature, Defined rule.

I. INTRODUCTION

In recent years, not only personal computers but also mobile devices having excellent image quality and performance have been widely spread, and various types of applications operating on such hardware have been developed. In particular, these devices have high-speed wired and wireless network functions and location-based sensors, thus the functions of the applications being developed are becoming more diverse [1-5]. Applications developed in this way run on a variety of platforms, and they are actually applied and used in various fields related to the Fourth Industrial Revolution, including artificial intelligence, autonomous driving, robots, drones, 3D printers, virtual reality and extended reality [6-10].

The process of developing such an application consists of a planning stage, a design stage, a development stage, a test and quality inspection stage, and a maintenance stage. Among them, the quality inspection stage of the developed application program usually checks whether all functions operate normally by hand, and directly checks whether the execution screen displayed in relation to the function is appropriate. However, in recent years, the number of applications being developed is so large that there is a limit that a person manually checks the quality of the application.

Therefore, there is a need for a study to automatically perform the quality inspection of the developed application rather than the manual work [11]. In other words, it is necessary to capture the screen displayed when each function of the corresponding application program is executed, and then determine whether the captured image is the same as the normal reference image through image processing.

Existing studies for detecting similar images from different types of images received can be found in the related literature. The research [12] proposed a method of analyzing the histogram of the image and confirming the distribution of the contrast and the color value of the entire image. Therefore, in this method, the histograms between two images are compared to calculate the similarity between the color values of the images. This method of comparing histograms has the advantage that the algorithm is relatively fast and the processing speed of the algorithm is fast.

The research [13] proposed a new method for finding a region corresponding to a target from an input image using template matching. In general, template matching has the advantage of solving a moving problem, but also has the disadvantage that it is difficult to match the object rotated or changed in size.

The research [14] proposed a method of judging whether high resolution satellite images are similar by matching feature points that are robust to changes in lighting. As described above, the feature point-based image matching method extracts features such as color, edge, line, and corner from two images, and then compares the extracted features with each other to calculate the dissimilarity of the images.

Manuscript published on January 30, 2020.
* 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

The research [15] proposed a method of retrieving similar images by measuring similarity based on the circular sector method. In this algorithm, the HSB color model is used for feature extraction, and the median function is used.

The proposed method in [15] has slightly higher accuracy than the existing method. In addition to the algorithm described above, new methods for accurately detecting normal and error images have been introduced [16].

However, the existing methods mentioned above are still relatively low in their degree of completion, and there are disadvantages in that there are many constraints depending on the surrounding environment in which the image is captured. Moreover, the research methods of detecting error images in connection with automated quality check of application program interface are relatively few compared to other existing research fields.

In this study, we describe a strategy that robustly detects faulty images that show incorrect interface screens, except for normal images captured by a properly working application program, by comparing and analyzing the major multiple features from the input images. Fig. 1 shows an overview of the error screen detection algorithm by comparing and analyzing the main features presented in this paper.

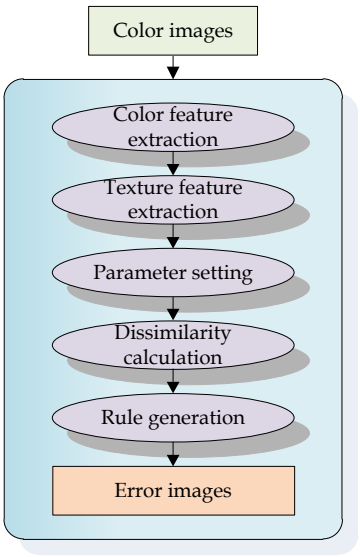


Fig. 1.Overall flow of the suggested method

As illustrated in Fig. 1, the introduced system extracts main multiple features representing the images from the captured images, which represent each screen on which the application program is executed. Subsequently, a difference operation is performed on the extracted feature, and it is checked whether the difference result is within a defined threshold. Through this process, it is effectively determined whether the input image is the same normal image as the reference image or an error image similar to the target image but different from the target image.

Chapter 1 describes the motivation, background, and outline of the study. Chapter 2 describes a technique to accurately extract the main features that represent the input image. Chapter 3 describes the technique of calculating the difference between the feature values and then determining whether the input image is a normal image or an error image using the calculated difference of the feature values. Chapter 4 explains the experimental results to quantitatively verify the introduced image judgment system. Section 5 describes the conclusions and future research plans.

II. FEATURE EXTRACTION

This chapter describes how to acquire the main characteristics that represent the input image. To this end, the image of 3 channels composed of RGB color space is first separated into each of R, G, and B channels, and then the difference feature for each color channel between the target image and the input test image is extracted.

In this paper, to check the color as well as the location and size of the objects in the image, the color channels are separated and the difference in color characteristics between the two images for each channel is obtained as shown in (1). In (1), $R_{ref}(x, y)$, $G_{ref}(x, y)$, and $B_{ref}(x, y)$ represent R, G, and B color values at (x, y) in the target image, respectively, and $R_{inp}(x, y)$, $G_{inp}(x, y)$, and $B_{inp}(x, y)$ represent R, G, and B color values at (x, y) in the input test image, respectively.

$$\begin{aligned} Diff_r(x,y) &= \left|R_{ref}(x,y) - R_{inp}(x,y)\right| \\ Diff_g(x,y) &= \left|G_{ref}(x,y) - G_{inp}(x,y)\right| \\ Diff_b(x,y) &= \left|B_{ref}(x,y) - B_{inp}(x,y)\right| \end{aligned} \tag{1}$$

The difference calculation result of each color channel generated above has a value between 0 and 255. If the result value of the difference operation is 0, it means that the pixel value of the target image and the corresponding pixel of the test image match. If the result value of the difference operation is 255, it means that the corresponding pixels do not completely match.

Equation (2) calculates the number of pixels whose color values are not similar for each channel by applying the difference calculation result value of (1) to all pixels of the input image. In (2), $Dissimil_r$, $Dissimil_g$, and $Dissimil_b$ represent the number of pixels whose R, G, and B color values are not similar to each other in the target image and the input image. TH_r, TH_g, and TH_b have a value between 0 and 255, and indicate thresholds of R, G, and B color values, which are the criteria for determining that two pixel values are not similar in the difference image [17-20]. In general, in the image processing field, the threshold value may be artificially set to an optimal value for a corresponding domain through repeated experiments.

$$\begin{aligned} &IF\ (Diff_r(x,y) > TH_r) \\ &\quad THEN\ Dissimil_r\ ++ \\ &IF\ (Diff_g(x,y) > TH_g) \\ &\quad THEN\ Dissimil_g\ ++ \\ &IF\ (Diff_b(x,y) > TH_b) \\ &\quad THEN\ Dissimil_b\ ++ \end{aligned} \tag{2}$$

In this paper, we use the texture feature as the second major feature to effectively measure the similarity between two images.

In an image, texture is generally used as one of the major matching features because it represents roughness of an image pixel distribution [21-24]. Conventional methods for extracting textures in the field of pattern recognition include many forms such as local binary pattern (LBP), wavelet transform, and so on. In this study, we utilize the Gabor feature [25], which is known to work relatively well in terms of accuracy. Gabor wavelets have the shape of plane waves constrained by Gaussian envelopes.

The Gabor feature used in this paper is expressed as in (3). Each wavelet $g_{mn}(x, y)$ may be made by the adequate dilation and rotation of the mother wavelet $g(x, y)$ via a generation function.

$$g(x,y)=\left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\times \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^{\,2}}+\frac{y^2}{\sigma_y^{\,2}}\right)+2\pi jWx\right) \quad (3)$$

$$g_{mn}(x,y)=a^{-m}g(x',y')$$

$$x'=a^{-m}(y\sin\theta+x\cos\theta)$$

$$y'=a^{-m}(y\cos\theta-x\sin\theta)$$

In (3), m and n denote expansion and rotation indices, respectively, $\theta = n\pi / K$, and K represents the number of rotations, which is set to 4 in this study.

The texture characteristic used in this paper is defined using Gabor wavelet coefficients calculated by convoluting the gray value of the pixel with the wavelet kernel, as shown in (4). In (4), $I(x, y)$ represents the gray value of the pixel at the (x, y) position of the image. G_{max} represents the maximum value of the texture feature. Therefore, the texture feature $G(x, y)$ is normalized to a value between 0 and 1. In addition, the calculated value represents a texture feature of the associated pixels.

$$G(x,y)=\frac{I(x,y)\cdot g_{mn}(x,y)}{G_{\max}} \quad (4)$$

Therefore, the multiple features used in the paper are defined by four main features, as shown in (5).

$$F=\left\{Dissimil_r, Dissimil_g, Dissimil_b, G(x,y)\right\} \quad (5)$$

III. DETECTION OF FAULTY IMAGES

In this paper, we use a metric as shown in (6) to measure the similarity of color features between the target image and the input test image. In (6), I_{ref} and I_{inp} represent the target image and the input image, respectively. W and H mean the horizontal length and the vertical length of the sample data, respectively. In (6), the color similarity is normalized to a value between 0 and 100 for ease of calculation.

$$C_{diff}(I_{ref},I_{inp})=\left(1-\frac{Dissimil_r+Dissimil_g+Dissimil_b}{3\times W\times H}\right)\times 100 \quad (6)$$

In this paper, we use the similarity measure as shown in (7). In (7), $G_{ref}(x, y)$ and $G_{inp}(x, y)$ denote the Gabor texture features extracted for the (x, y) positions of the target image and the input test image.

$$T_{diff}(I_{ref},I_{inp})=\left(\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(1-\left|G_{ref}(x,y)-G_{inp}(x,y)\right|\right)}{W\times H}\right)\times 100 \quad (7)$$

In this paper, we define the similarity measure like (8) to calculate the final similarity between a target image and a test image by integrating similarity measure of color feature and texture feature. In (8), α and β are the weighting factors [26] that represent the contribution of color and texture features. In this paper, the weighting factor has a value between 0 and 1, and the sum of all weighting factors is 1. In this paper, the values of α and β are set equal to 0.5, respectively. In (8), the similarity measure $Simil(I_{ref}, I_{inp})$ has a value between 0 and 100. Therefore, if the value of the similarity measure is 100, the target image and the input image mean the same image, and if the value of the similarity measure is 0, the images are not completely similar.

$$Simil(I_{ref},I_{inp})=\alpha\times C_{diff}(I_{ref},I_{inp})+\beta\times T_{diff}(I_{ref},I_{inp}) \quad (8)$$
$$where\ 0\le\alpha,\beta\le 1,\ \alpha+\beta=1$$

Finally, in this paper, the error image is automatically detected from the input test image using the rule [27] defined as in (9). In (9), TH_{error} is a threshold used to determine whether two images are similar or not.

$$\begin{aligned} &IF\ (Simil(I_{ref},I_{inp})\ge TH_{err})\\ &\quad THEN\ I_{ref}\ and\ I_{inp}\ are\ similar\\ &ELSE\ (Simil(I_{ref},I_{inp})<TH_{err})\\ &\quad THEN\ I_{ref}\ and\ I_{inp}\ are\ dissimilar \end{aligned} \quad (9)$$

In other words, if the similarity value $Simil(I_{ref}, I_{inp})$ between the two images is greater than or equal to the threshold TH_{err}, the input color image is determined to be a normal image similar to the reference image. Otherwise, the input color image is determined to be an error image that is dissimilar to the reference image.

Fig. 2 shows the overall flow diagram of detecting error images from input test image.

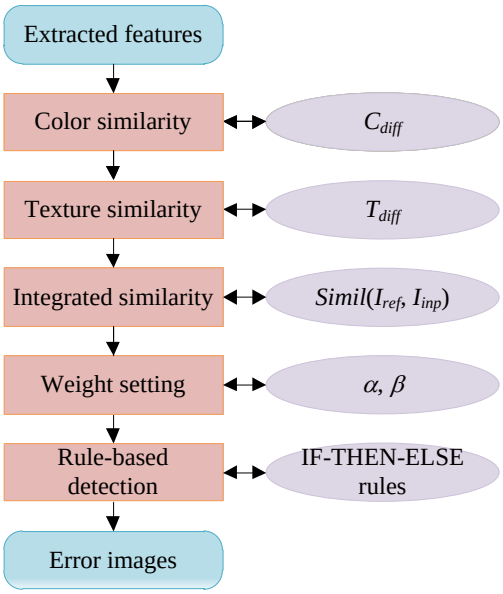


Fig. 2.Flow diagram of detecting error images

IV. EXPERIMENTAL RESULTS

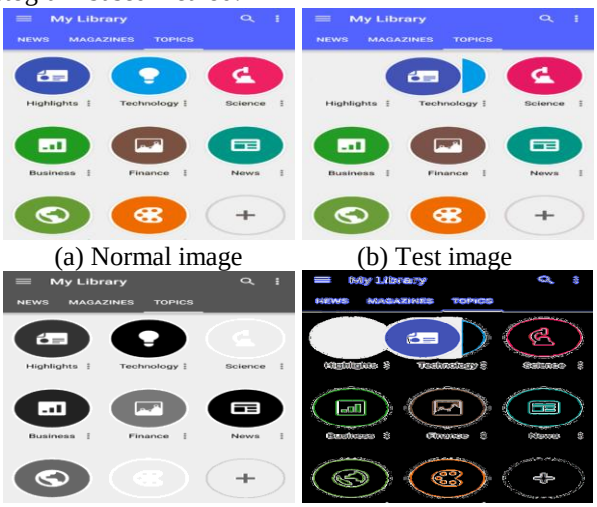
The personal computer utilized to perform the experiments in this study is composed of an Intel Core (TM) i7-6700 3.4 GHz CPU, 16 GB of main memory, 256 GB of SSD, and a Galaxy Geforce GTX 1080 Ti graphics card with NVIDIA's GPU GP104. The personal computer used was installed with the Microsoft Windows 10 operating system (OS). In addition, Microsoft Visual Studio version 2015 was used as the Integrated Development Environment (IDE), and the proposed algorithm was developed using the OpenCV version 3.4.7 Open Vision library. In the experiment, each execution screen of the developed application was captured and used as test data.

Fig. 3 (a) shows an example of an input color image that normally shows the execution result screen of the application program used in this paper. Fig. 3 (b) is a test image capturing an execution screen of an application program, and shows a test image in which an error that is not normally executed occurs. Figure 3 (c) illustrates an example of the color characteristics of the R channel extracted from the input image. Fig. 3 (d) shows the dissimilar regions detected from the target image and the test image by analyzing the color and texture features.

In this paper, we quantitatively evaluate and compare the performance of the proposed dissimilar region extraction algorithm. In this paper, we used a performance measure such as (10). In (10), $Num_{detected}$ indicates the number of dissimilar images that are correctly detected using the proposed approach. Num_{total} represents the total number of dissimilar images existing in the input image data.

$$\Phi_{measure} = \frac{Num_{detected}}{Num_{total}} \times 100\ (\%) \qquad (10)$$

Table I gives the results of the performance measurement in terms of the accuracy of the faulty image detection method in a tabular form. In this study, we compared the performance of the conventional histogram-based method and the existing matching-based method with the proposed method. As given in Table I, it can be seen that the algorithm presented in this paper detects faulty images of the execution result of the application program more accurately than the conventional methods. In addition, the performance of the conventional histogram-based method was the worst, and the conventional matching-based method was worse than the proposed method, but the performance was somewhat better than the histogram-based method.



(a) Normal image (b) Test image

(c) Red component image (d) faulty region

Fig. 3.Faulty region detection

The conventional histogram-based method has the advantage that the algorithm is relatively simple and fast. However, this method generates an error of judging the same scene even if the position of the target object area is changed. In the conventional matching-based method, when the size of the object is changed or the object is rotated, the matching is inaccurate and an error occurs. On the other hand, the proposed method is more accurate than the conventional methods by effectively extracting the difference between the color and texture features between two images.

Table I. Performance comparison

Method	Accuracy (%)
Conventional histogram-based method	87.13 %
Conventional matching-based method	87.57 %
Proposed method	89.92 %

Besides, the existing methods of extracting similarity between images based on single feature or histogram comparison mostly lack information to sufficiently express the entire image. Therefore, when similar areas between two images are extracted, many errors occurred due to incorrect matching. However, the proposed multiple feature-based method obtains richer information by extracting multiple features from the input color image. In addition, because the matching is done based on this information, the accuracy is relatively high.

V. CONCLUSION

In recent years, the number of applications to be developed increases exponentially, which makes it difficult to manually inspect the quality of the developed screen. Therefore, it is necessary to study the quality by automatically analyzing the adequacy of the execution screen of an application program using image processing techniques.

In this study, we describe a strategy to accurately detect the execution error screen of the developed application program by performing multiple feature analysis from various types of test color images. The proposed method first extracts multiple features that best represent the input color image. Then, by comparing and analyzing the extracted multiple features, it is effectively determined whether the input image is a normal image similar to the target image or an error image similar to the reference image but different from each other. Experimental results give that the algorithm presented in this study detects the execution error result screen of the application program more accurately than the existing methods from the input test image.

In the future, we will apply the dissimilar region extraction algorithm presented in this paper to the execution result screen of various types of application programs to confirm the performance of the introduced strategy in terms of accuracy and time. In addition, we plan to upgrade the scalability and compatibility of the introduced error scene detection algorithm by applying it to various other platforms as well as personal computers and mobile devices.

REFERENCES

1. M. Abdullah, W. Iqbal, A. Erradi, "Unsupervised learning approach for web application auto-decomposition into microservices," J. Syst. Software, vol. 151, 2019, pp. 243-257.
2. J. Soldani, T. Binz, U. Breitenbucher, F. Leymann, A. Brogi, "ToscaMart: a method for adapting and reusing cloud applications," J. Syst. Software, vol. 113, 2019, pp. 395-406.
3. C. S. Garcia, A. Meincheim, E. R. F. Junior, M. R. Dallagassa, E. E. Scalabrin, "Process mining techniques and applications - A systematic mapping study," Expert Syst. Appl., vol. 133, 2019, pp. 260-295.
4. J. Liang, J. Gong, W. Li, "Applications and impacts of Google Earth: A decadal review (2006-2016)," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 146, 2018, pp. 91-107.
5. S. Han, Y. Yun, E. Hwang, W. Kim, Y. H. Kim, "Event-based application modeling for analysis of asymmetric multicore-based mobile systems," J. Syst. Architect., vol. 97, 2019, pp. 477-490.
6. L. L. Iacono, P. L. Gorski, J. Grosse, N. Gruschka, "Signalling over-privileged mobile applications using passive security indicators," Journal of Information Security and Applications, vol. 34, 2017, pp.27-33.
7. A. Chetouani, S. Treuillet, M. Exbrayat, S. Jesset, "Classification of engraved pottery sherds mixing deep-learning features by compact bilinear pooling," Pattern Recogn. Lett., vol. 131, 2020, pp. 1-7.
8. M. Ianni, E. Masciari, G. M. Mazzeo, M. Mezzanzanica, C. Zaniolo, "Fast and effective big data exploration by clustering," Future Generat. Comput. Syst., vol. 102, 2020, pp. 84-94
9. A. Yang, Y. Li, C. Liu, J. Li, J. Wang, "Research on logistics supply chain of iron and steel enterprises based on block chain technology," Future Generat. Comput. Syst., vol. 101, 2019, pp. 635-645.
10. S. HUANG, D. QI, J. YUAN, H. TU, "Review of studies on target acquisition in virtual reality based on the crossing paradigm," Virtual Reality and Intelligent Hardware, vol. 1, no. 3, 2019, pp. 251-264.
11. M. Babu, P. Franciosa, D. Ceglarek, "Spatio-temporal adaptive sampling for effective coverage measurement planning during quality inspection of free form surfaces using robotic 3D optical scanner," J. Manuf. Syst., vol. 53, 2019, pp.93-108.
12. S. Borjigin, P. K. Sahoo, "Color image segmentation based on multi-level Tsallis–Havrda–Charvat entropy and 2D histogram using PSO algorithms," Pattern Recogn., vol. 92, 2019, pp. 107-118.
13. E. Elboher, M. Werman, "Asymmetric correlation: a noise robust similarity measure for template matching," IEEE Trans. Image Process., vol. 22, 2013, pp. 3062-3073.
14. B. Wu, H. Zeng, H. Hu, "Illumination invariant feature point matching for high-resolution planetary remote sensing images," Planet. Space Sci., vol. 152, 2018, pp. 45-54.
15. J. Masek, R. Burget, L. Povoda, and M. Harvanek, "Image search using similarity measures based on circular sectors," in Proceedings of the Fourth International Conference on Advanced Information Technologies and Applications, Harbin, China, 2015, pp.241-251.
16. J. Shi, X. Wang, "A local feature with multiple line descriptors and its speeded-up matching algorithm," Comput. Vis. Image Understand., vol. 162, 2017, pp.57-70
17. J. Liang, D. Liu, "A local thresholding approach to flood water delineation using Sentinel-1 SAR imagery," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 159, 2020, pp.53-62.
18. 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.
19. 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.
20. 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.
21. P. Banerjee, A. K. Bhunia, A. Bhattacharyya, P. P. Roy, S. Murala, "Local neighborhood intensity pattern - a new texture feature descriptor for image retrieval," Expert Syst. Appl., vol.113, 2018, pp.100-115.
22. J. J. M. Junior, L. C. Ribas, O. M. Bruno, "Randomized neural network based signature for dynamic texture classification," Expert Syst. Appl., vol. 135, 2019, pp. 194-200.
23. M. A. Mizher, R. Sulaiman, A. M. Abdalla, M. A. Mizher, "An improved simple flexible cryptosystem for 3D objects with texture maps and 2D images," Journal of Information Security and Applications, vol. 47, 2019, pp. 390-409.
24. H. Al-Marzouqi, Y. Hu, G. AlRegib, "Texture retrieval using periodically extended and adaptive curvelets," Signal Process. Image Comm., vol. 76, 2019, pp. 252-260.
25. F. Riaz, A. Hassan, S. Rehman, U. Qamar, "Texture classification Using rotation- and scale-invariant Gabor texture features," IEEE Signal Process. Lett., vol. 20, no. 6, 2013, pp. 607-610.
26. Y. Yu, H. Zhao, "Novel sign subband adaptive filter algorithms with individual weighting factors," Signal Process., vol. 122, 2016, pp. 14-23.
27. O.-J. Lee, J. E. Jung, "Sequence clustering-based automated rule generation for adaptive complex event processing," Future Generat. Comput. Syst., vol. 66, 2017, pp. 100-109.

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. Since March 2009, he has been a Professor in the Department of Software, Anyang University, Korea. His research interests include artificial intelligence, block chain, Target object blocking, video surveillance, and adult image detection.

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 has been a Professor in the Department of Computer Engineering at Anyang University, Korea. His primary research interests include neuro-fuzzy systems, computer vision, and big data prediction system.