Automatic Recognition of Flowers through Color and Edge Based Contour Detection

Soon-Won Hong and Lynn Choi
School of Electrical Engineering, Korea University
e-mail: aldig@korea.ac.kr, lchoi@korea.ac.kr

***Abstract*—Unlike simple images processed by the existing image-based search engines, flowers have wider and more irregular range of shapes and patterns. In this paper we present an automatic recognition system of flowers for smartphone users. After a user transmits a flower image to the server, the image processing and searching is performed only by the server, eliminating the user interaction from the recognition process. The server detects the contour of a flower image by using both color-based and edge-based contour detection. Then, we classify its color groups and contour shapes by using k-means clustering and history matching. After comparing the input image with the reference images stored on the server, the server sends the most similar image to the user. We also address the image recognition failure issue caused by the light and the camera angle by partial recognition and image recovery. We have obtained the success rate of 94.8 % for 500 images from 100 species.**

***Keywords*—flower recognition, contour detection, histogram matching, search engines, smartphones.**

I. INTRODUCTION

Unlike desktop computing, smartphones and tablet PCs provide the environment whenever and wherever people have access to various sensor-based information services such as GPS-enabled navigation and e-healthcare services. As the image recognition technology matures, the development of searching applications based on camera vision is now widely being adopted.

Image recognition technology[1, 2, 3, 4, 5] is used in diverse application areas such as medical equipment, automobiles, electronics, securities and distribution systems as the camera sensor and the image processing technology matures. Recently, the image recognition has extended its application to content based retrieval technique, which is quite different from the established text based retrieval. But, the retrieval by image recognition is still in a stage where only the subjects with simple shapes or similar patterns can be searched. Examples include characters, posters, or personages. To extend the range of images for the image-based search technology, more complex and irregular image patterns of various objects must also be recognized. In addition, images taken in a certain location are influenced by surroundings, such as light or camera angle, it is hard to get the correct images. In order to extend the application range of image based retrieval, the technology is needed by which all of various shaped subjects and images taken in any places can be recognized.

This study aims to extend the image-based search engine technology to automatic recognition of natural flowers. Flowers live in a natural environment where they are affected by the environmental factors such as sun light and surrounding objects. A few studies [6, 7, 8, 9] have investigated the possibility of the flower recognition. Saitoh [6] proposed a recognition system for wild flowers using flower contour. Zou [7] developed a system called Computer Assisted Visual InterActive Recognition (CAVIAR) and compared the results with those of human recognition. Ye [8] introduced a plant recognition web application based on text and content. Nilsback [9] proposed flower classification using a visual vocabulary that is composed flower features. Bosch [10] proposed Image Classification using Random Forests and compared Multi-way SVM with region of interst(ROI), visual word, and edge distributions. Kim [11] proposed a mobile-based flower recognition system using difference image entropy (DIE) and contour features. Boykov [12, 13] proposed a technique for general purpose interactive segmentation of N-dimensional images and demonstrated.

These studies only use contour information to reduce candidates, but we use both contour and color information to reduce candidates efficiently. Also, several studies [7, 11] require a user to manually adjust the outline of flowers on device. Although this approach can extract flow contour more exactly, it is neither user friendly nor a fully automatic recognition process. Therefore, in this study we eliminate the user interaction form the flower recognition process. Also, this study uses flower images of natural conditions as inputs, not secondary images that eliminate or minimize the influence of environmental factors such as light, camera angle or surrounding objects. After a client transmits a flower image to the server, the image processing and searching is handled only by the server. First, the server reduces noises, normalizes the image pattern and eliminates unnecessary backgrounds. After the server detects the contour of a flower image through color and edge-based contour detection, it extracts its color and distance information of contour. Then, it compares these image features with the features of flower images stored in the server's databases through k-means color clustering and distance histogram matching. In addition, we perform partial recognition and image recovery to reduce the influence of light, and camera angle, improving the recognition rate. We have performed the experiments on 500 images taken from 100 species of Korean natural flowers, and obtained the recognition success rate of 85.2%, 89.8%, and 94.8% for the first, second, and third-rank accumulated respectively. Our results confirm that we can not only increase the recognition success rate but also increase the number of species substantially compared to the existing studies.

This paper is organized as follows. Section 2 presents the overall architecture of our flower recognition system. Section 3 describes the experimentation environment and analyzes the results. Section 4 concludes the paper.

978-1-4673-2584-4/12/$31.00 ©2012 IEEE

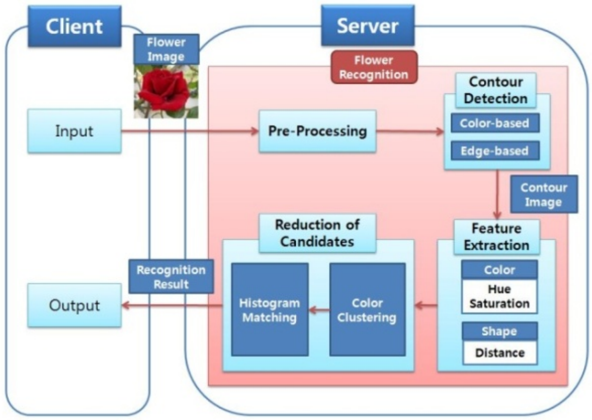


Fig. 1. The overall system architecture

II. Flower recognition system architecture

Figure 1 shows the overall system architecture of the proposed automatic flower recognition system. The client user acquires flower images using his or her smartphone and sends the input images to the server where our flower recognition system is located. All the flower recognition process is handled by the server. The server finds the contour of flowers using color and edge information, and then extracts the image features of flowers. The server compares these features with the features of images stored in the server through color clustering and distance histogram matching. Then, the system determines the species of the flower that has the most similar features. Since all the image processing and searching is performed by the server, we can eliminate the user interaction from the recognition process except acquiring and transmitting camera images to the server.

A. Pre-Processing

To remove the noise from the image, the system performs smoothing and morphological transformation in pre-processing. To acquire a smoother image, we use Median Filter [14] that selects median value around each pixel.

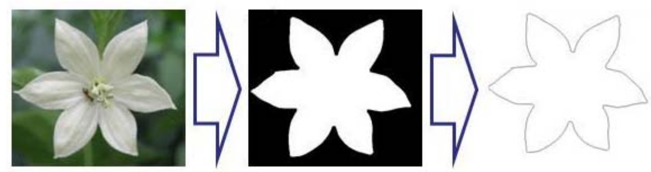
Basically, the morphological transformation [15] is a dilation and erosion operation. The erosion operation can remove the small chunks of objects depending on the filter size and the number of uses. Also, the dilation operation can remove the small holes from the object image.

To eliminate the interference from the background in the image processing, we also eliminate unnecessary background from the image and leave only flower contours. The background is transformed into black background to contrast the flower colors, which we classify into 5 basic color groups: red, yellow, blue, pink, and white. When the system detects an outline of contours, it also removes all the other color groups except the main color group to remove the interference from other color groups. Why we classify into 5 basic color groups are described in Section III.

B. Contour Detection

In general, flowers in a natural state possess colors that stand out more than surroundings. In other words, it is easy to detect the contour of a flower by using color information. Although colors show distinctive difference to the naked eye, color data collected from an image may hold redundant or sometimes ambiguous information. Instead of relying only on color information, we use both color-based and edge-based contour detection to further improve the quality of the contour detection. The flower contour can be detected by using either one of the methods. However, it is not always possible to obtain the contour depending on species of flower. By using both color-based and edge-based contour detection, we can improve the contour detection performance. We first apply color-based contour detection. If the contour is not found, the flower is recognized through edge-based contour detection. This improves the contour detection accuracy since either edge-based or color-based contour detection alone may fail. Why we select color-based contour detection as our main contour detection is described in Section III.

1) Color-based Contour Detection: In this paper, flowers in an image are assumed to be located in the middle. A background-eliminated image performs masking on the input image. Then, the color-based contour detection starts at masking point, which is usually located in the floral leaf since the center image often contains pistils which may have different colors. If the part of floral leaf is detected, it is updated to a new contour. As shown in Figure 3, we find the contour of a flower by comparing the hue value of reference pixels to nearby pixels (3x3) since the background pixels have similarity in colors, which are usually different from the flower color group.



(a) (b) (c)

Fig. 2. Color-based contour detection process. (a) input, (b) color-based detection, (c) detected contour

Figure 2 shows color-based contour detection process. The final contour is obtained by assuming the biggest contour to be a flower and by removing inner-contour.

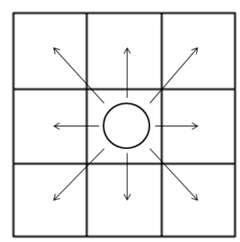


Fig. 3. The hue value of reference pixels (include circle)

Figure 3 shows the color-based contour detection process in detail. The hue value of reference pixels are compared to hue value of 3x3 nearby pixels. If the difference value is less than 5, the pixel is decided to a part of floral leaf. This process is repeated for the entire pixels. If the difference value is more than 5, the process is stopped and last pixel is decided to contour of flower.

2) Edge-based Contour Detection: Flowers may have several central pistils and stamens. And, the color of petal is often different from the color of pistils and stamens. When the color of an image's center is clearly different from the color of the flower leaves, it will be difficult to detect a flower contour. In this case its contour is detected by finding the edge of a flower image. Finding an edge is equivalent to finding a point whose image intensity undergoes a rapid change. This may be seen as finding a point with a great image intensity slope, and may be derived through first derivation. First derivative's quadratic function f(x, y) is expressed by the following equation.

$$\nabla f(x,y) = \frac{\partial f(x,y)}{\partial x} i_x + \frac{\partial f(x,y)}{\partial y} i_y \qquad (1)$$

where i_x, i_y are x direction, and y direction's unit vector.

Sobel Differential Operation [16], the most popular derivative operation, is used to derive a differential value. If the intensity change occurs gradually over the course of a wide area, a thick contour may be detected as a slope-type contour. Therefore, our system employs Canny Edge Detector [17]. The most distinctive feature of Canny Edge Detector is that it collects candidates for possible edge pixels as forms of contour by using 2 critical values. If an edge's intensity is greater than a higher critical value, it is designated as an edge pixel; and if the intensity is less than a lower critical value, it is excluded from an edge. The edges are acknowledged on each pixel only when the intensity exhibits values between two critical values and when the edge pixels exist around the pixel.

(a) (b) (c)

Fig. 4. Edge-based contour detection process. (a) input, (b) edge-based detection, (c) detected contour

Figure 4 shows all the edges detected by Canny Edge Detector. The final contour is obtained by assuming the biggest contour to be a flower and by removing interferences.

C. Feature Extraction

We convert an input image given in RGB color into HSV (hue, saturation and value) domain [18]. HSV rearranges the geometry of RGB color in an attempt to be more intuitive and perceptually relevant. In addition, we find that the HSV representation can better tolerate the color distortion caused by rays and shadows than the original RGB representation. In addition, HSV is more useful in color clustering since HSV is amenable to human perception. Hue represents color information with angular dimension, starting at red color at 0°, passing through green at 120° and blue at 240°, and then wrapping back to red at 360°. Saturation represents purity in color, considering black or white as full saturation of 0 while pure color such as red, yellow, or blue as saturation of 1. Value represents brightness where white gives 100% while black gives 0%. Figure 5 plots hue, saturation, and value in cone representation. As value decreases, the impact of saturation on hue is also decreased. Therefore, we can ignore the value, obtaining the color information by using only hue and saturation.

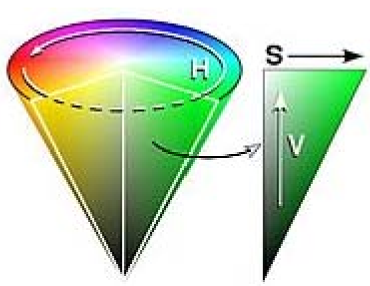


Fig. 5. HSV color space

Figure 6 shows the flower shape analysis process, where the edge is displayed in the two-dimensional space by moving the contour image counter-clockwise starting from 12'o clock by computing the distance value (D) between the center and the contour edge. Distance values acquired above are saved in a histogram bin from point of longest distance value, which are compared to histogram values previously saved in the server.

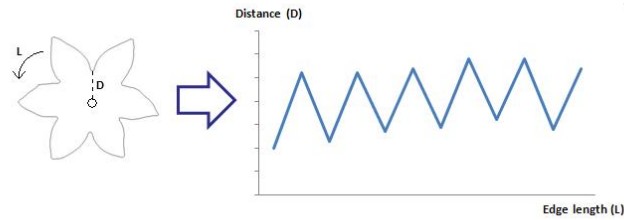


Fig. 6. Flower shape analysis

D. Reduction of candidates

The exiting studies [6, 7, 9, 11] mainly rely on contour feature for flower recognition. However, since a flower species may have different colors, it is not easy to differentiate flowers with different colors based only on contour feature. Thus, in this study we use both color and contour features to improve the recognition quality. Specifically, we use color clustering and distance histogram matching to reduce the number of candidates and to further improve the recognition success rate.

1) Color Clustering: Flowers are largely classified into 5 color groups, but it is not easy to clearly delineate a range for each color group since a color group consists of contiguous values of hue (0~360°), saturation (0~100%), and intensity (0~100%). To overcome this issue we conduct two-dimensional clustering by using hue and saturation values. We use k-means clustering because it has a simple structure and operates rapidly in various settings [19, 20]. Pixel hue and saturation values are disintegrated into 5 groups and all the data points (hue, saturation) are assigned to the center of the closest group by using Euclid distance calculation. Next, the center of a new group is calculated by minimizing the distance between each data point assigned to the group and the center of the group. This process is repeated until there is almost no change in the center of the group. When a new flower image is collected, the groups are updated including the new color information. K-mean clustering has outlier problems. However, we have already known that what group is with color based contour detection. So, we can eliminate outliers.

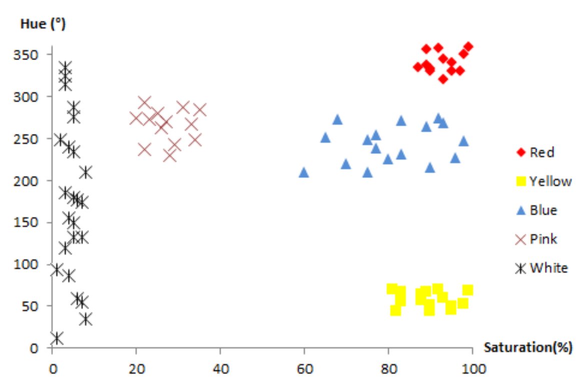


Fig. 7. Generation of 5 color groups through color clustering

Figure 7 shows 5 color groups classified by the color clustering for our input images tested in Section III.

2) Distance Histogram matching: To measure the similarity of an input image and a reference image, we use a technique called history matching. We compare distance values in a histogram bin to histogram values of reference images previously saved in the server. Comparing 2 histogram values using the likelihood function was first introduced by Swain and Ballard [21], and upgraded by Schiele and Crowley [22]. We also use the crossover method to measure the similarity of histogram values. The following equation expresses the distance values where H_1 and H_2 represent the histogram of an input image and the histogram of a reference image respectively.

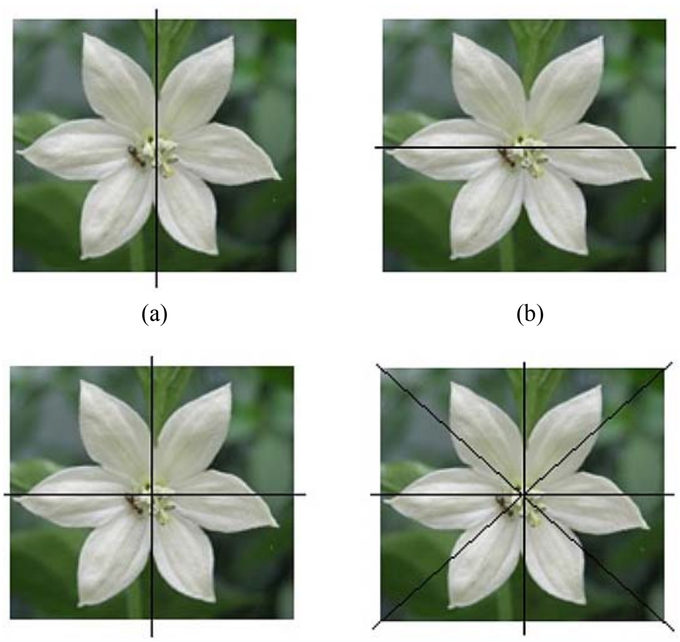
$$d_{intersection}(H_1, H_2) = \sum_i \min(H_1(i), H_2(i)) \qquad (2)$$

After two histograms are regularized, if Equation (2) leads to a value closer to 1, it implies that two images are close to identical. If it shows a value closer to 0, two histograms have nothing in common.

3) Partial Recognition: The image in outdoor cannot avoid the effect of light. This means that the color of image can be changed unexpectedly from the original color by the light, causing a completely different result. Figure 8 shows the examples, which actually lead to incorrect recognition results in our contour detection.

(a) (b)

Fig. 8 (a) A watermelon flower example affected by light, (b) A tulip example affected by light



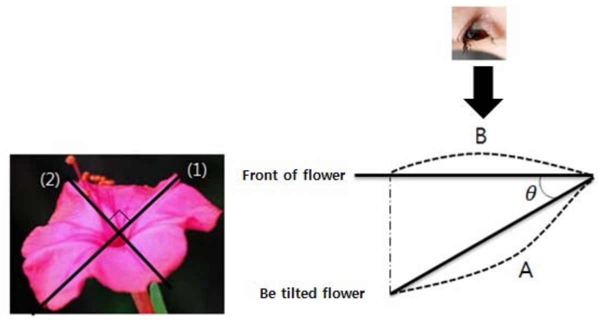
(a) (b)

(c) (d)

Fig. 9. An image (a) divided by 2 as vertically, (b) divided by 2 as horizontally, (c) divided by 4, and (d) divided by 8

To solve this problem, this study proposes partial recognition. As shown in the Figure 9, the image was divided by 2, 4 and 8 sections. The image can be split into two directions vertically or horizontally, so total five types of images are compared including the original input image. By summarizing the result with the result of divided images, the flower with the most matched shall be the final result. In general, as the image is divided more, the result of affected part by the light can be removed. However, a wrong result may come out because the information of contour to be compared becomes less if the image is divided too much. So the image should be divided in a proper number. In our experimentation, dividing the image by 4 has the highest recognition rate compared to dividing the image by 8.

4) Consideration of Flower Angle: The same flower may look differently depending on a camera angle, which may cause incorrect contour and incorrect recognition result. The best solution is to have the flower images from every angle in the server database. Since this requires a large amount of images, it may not be an efficient solution. This study suggests the solution on the matter of angles by using the images from the several angles of 30, 45, 60 degrees, etc. not from every angle.



(a) (b)

Fig. 10 (a) An image of a lying down garden balsam (b) Thick lines mean cross section of the flower. To calculate theta, trigonometrical function is used.

It is assumed that the flower image is round and the diameter of the flower is already known. As seen in Figure 10 (a), (1) the figure is out in diameter of the longest flower and (2) draw a line crossing the right angle. Figure 10 (b) shows the images of the flower taken from the front and taken in the state of lying down. The length of the line (2) in Figure 10 (a) and the line B in Figure 10 (b) is the same. And since the length of A is the diameter of the flower seen from the front, the following mathematical formula is established.

$$\cos\theta = \frac{B}{A}$$

And theta can be represented like the following.

$$\cos^{-1}\frac{B}{A} = \theta$$

After finding the value of theta, the matching process can proceed using only the most similar images with adjacent angles.

III. Experiments and Results

To evaluate the performance of our proposed flower recognition process, we have implemented our automatic flower recognition system by using Microsoft Visual Studio 2008. As a result of the recognition, the recognized flower species were ranked up to the third rank. Figure 11 shows a recognition system interface, which shows top 3 image candidates for an input image. Table 1. Flower color group distribution of test images

Flower Color Group	# of Images
Red	90
Yellow	130
Blue	60
Pink	70
White	150
Total	500

100 most popular flower species that can be easily found in Korea

are selected as test species, from which 5 images for each species are collected. Therefore, 500 JPG images are used as test inputs. Table 1 shows the distribution of 500 images by categorizing into 5 flower color groups.

The color of flower is normally classified in 4~5 color groups in ordinary flower dictionary. In this study, it is classified into 5 color groups. As various colors of even the same flower may exist, we can classify the color groups in more detail as more images are additionally collected.

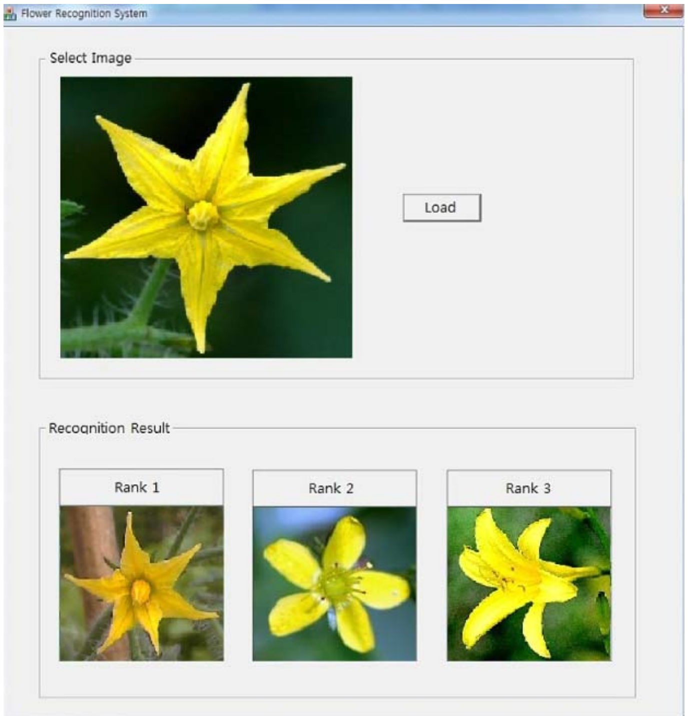


Fig. 11. Flower recognition system interface

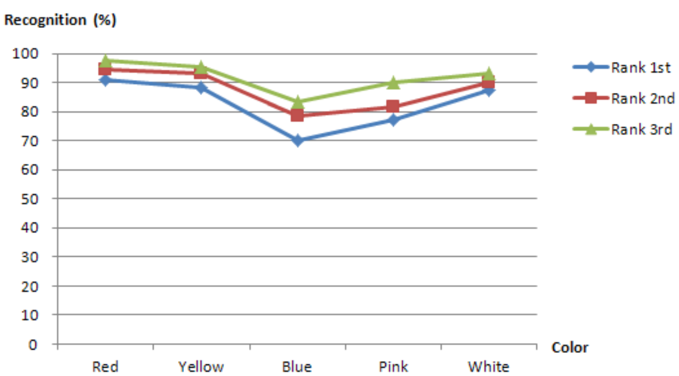


Fig. 12. Recognition success rate for five color groups

For the evaluation of the recognition rate, k-fold cross validation, one of the most generally used evaluation methods in machine learning, is used. 100 images were divided into 5 sub-sets, among which 100 are used as test sets and the other 400 as training sets. An average value was derived from the result of 5-time recognition trials. Figure 12 shows the recognition success rate for five color groups. 1st, 2nd, and 3rd-rank result show the accumulated recognition success rate for the first candidate, for the first and the second candidates, and for the first, second, and third candidates respectively. 3rd-ranked candidates show the highest recognition rate, since they represent an accumulated value from the 1st to 3rd ranks. A comprehensive recognition rate for 500 flowers is as follows: 85.2% for the first rank, 89.8% for the second rank, and 94.8% for the third rank. Blue and pink flowers recognition rate is lower than other color groups, since blue and pink flowers are relatively fewer in numbers.

Figure 13 shows the recognition rate for the third rank by applying different contour detection procedures. As shown in the figure, the color-based contour detection yields a higher recognition rate than the edge-based contour detection since a small number of color groups facilitate the flower classification process. As shown in the figure, applying both color-based and edge-based contour detection yields higher recognition rates than applying either color-based or edge-based contour detection alone. This shows that we need to reduce the number of candidates by applying color-based contour detection before we apply the additional edge-based contour detection to obtain the best result.

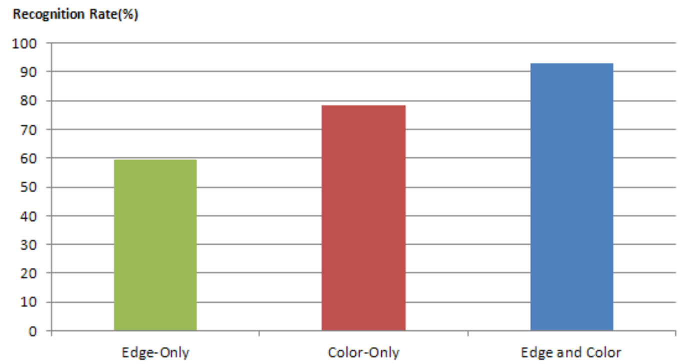


Fig. 13. Recognition result according to different contour detection procedures

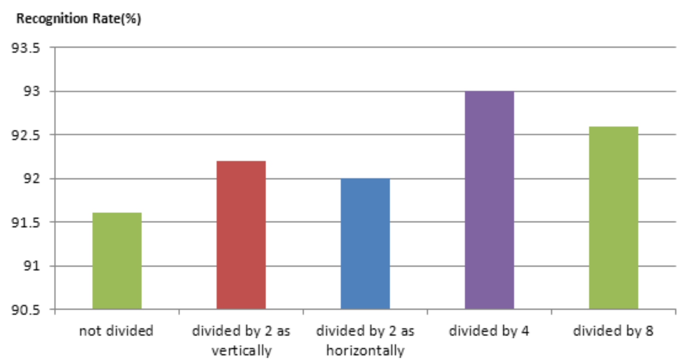


Fig. 14. Recognition result according to number of divided

Figure 14 shows the recognition rate according to the number of divided images. As the image is more divided, we can eliminate the effect of light, improving the recognition rate. However, the result of image divided by 8 is lower than that of image divided 4. This means that the more the image is divided, the less the contour information remains. As shown in Figure 14, in our experimentation the dividing the image by 4 shows the highest recognition rate.

Table 2 shows the number of the flowers that failed to be recognized. There were three main reasons for that. Totally 33flowers failed to be detected - 12 of them are light Influence, three are from the existence of similar color in the image background and 18 were from the angle of flowers.

First, we can reduce light influence from 7 images using partial recognition. Second case happens as the color of the flower and that of the background is similar or as the wrong contour is detected from another flower in the background. Finally, it's a matter of camera angle. So far, we could eliminate the angle problem from 1 example. We can apply the similar technique to eliminate the other angle cases, but this remains our future work.

Table 2. Failure analysis and the result of partial recognition and angle correction

Cause	# of images	# of images (after partial recognition and camera angle correction)
Light Influence	12	5
Existence of similar color objects in the image background	3	3
Angle Problem	18	17
# images correctly recognized	467	475
Total images processed	500	500

IV. Conclusions

In this paper we study a new image-based search system for flowers through color and edge based contour detection as well as color clustering. We also apply partial recognition and image recovery to reduce the influence of light or camera angle issue. Unlike the previous studies, we completely remove the user interaction from the contour detection and all the image processing and searching is performed by the server automatically. This study considers the best convenience of users because they aren't involved in the process except taking pictures. And, our flower recognition system can be applied to natural flowers images directly. When a new image of flowers is detected through color clustering, a better result can be produced next time by updating the established color information into our image database. To apply this technology to real life rather than experimentation labs, we have developed the server-client structured system that can be operated on smartphones. In addition, we increase the number of flower species processed by the system compared to the existing studies.

Since flowers in nature hold similar colors and shapes, it is impossible to distinguish all sorts of flowers only by flower colors or contours. Since even the identical flowers look different according to the angles, the contour of them can be extracted into the various forms. In this study we use both color-based and edge-based contour detection to improve the accuracy of the contour detection.

In our future work, we plan to apply data mining and machine learning technology to self-extract and learn the features of images taken by different users. We also expect the use of more sensor data to collect new features of flowers other than color and contour. This would include location and time, which can further improve the accuracy of searching.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012- 0002515).

References

[1] Ana Barman, Amit Kumar Samanta, Tai-hoon Kim and Debnath Bhattacharyya, "Design of a view based approach for Bengali Character recognition", International Journal of Advanced Science and Technology, vol. 15, pp.49-62, 2010

[2] Velu C M, P.Vivekanadan, Kashwan K R, "Indian Coin Recognition and Sum Counting System of Image Data Mining Using Artificial Neural Networks", International Journal of Advanced Science and Technology, vol. 31, pp.67-80, 2011

[3] Youssef Es Saady, Ali Rachidi, Mostafa El Yassa, Driss Mammass, "Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character", International Journal of Advanced Science and Technology, vol. 33, pp.33-50, 2011

[4] Mukesh Kumar, Anand Chauhan, Pankaj Kumar, "Handwritten Arabic Character Recognition: Which Feature Extraction Method?", International Journal of Advanced Science and Technology, vol. 34, pp.1-8, 2011

[5] K. Jaya Priya and R.S Rajesh, "A Local Min-Max Binary Pattern Based Face Recognition Using Single Sample per Class", International Journal of Advanced Science and Technology, vol. 36, pp.41-50, 2011

[6] Takeshi Saitoh, Kimiya Aoki, Toyohisa Kaneko, "Automatic Recognition of Blooming Flowers", in Proc. IEEE ICPR 2004, vol. 1, pp.27-30, 2004

[7] Jie Zou, and George Nagy, in Proc. the Pattern Recognition, "Evaluation of Model-Based Interactive Flower Recognition", IEEE ICPR2004, vol. 2, 2004, pp.311-314, 2004

[8] Yanhua Ye, Chun Chen, Chun-Tak Li, Hong Fu, and Zheru Chi, "A Computerized Plant Species Recognition System", International Symposium on Intelligent Multimedia, Video and Speech Processing, ISIMP 2004, pp.723-726, 2004

[9] Maria-Elena Nilsback, Andrew Zisserman, "A Visual Vocabulary for Flower Classification", IEEE Computer Society Conference on Computrer Vision and Pattern Recognition 2006(CVPR 2006), pp. 1447-1454, 2006

[10] Anna Bosch, Andrew Zisserman, Xavier Munoz, "Image Classification using Random Forests and Ferns" , IEEE International Conference on Computer Vision 2007(ICCV 2007), pp. 1-8, 2007

[11] Jung-Hyun Kim, Rong-Guo Huang, Sang-Hyeon Jin and Kwang-seok Hong, "Mobile-Based Flower Recognition System", Intelligent Information Technology Application, IITA 2009, vol. 3, pp.580-583, 2009

[12] Yuri Y. Boykov, Marie-Pierre Jolly, "Interactive Graph Cuts for Optimal Boudary & Region Segmentation of Objects in N-D Images", IEEE International Conference on Computer Vision 2001(ICCV 2001), pp. 105-112, 2001

[13] Yuri Y. Boykov, Marie-Pierre Jolly, "Demonstration of Segmentation with Interactive Graph Cuts", IEEE International Conference on Computer Vision 2001(ICCV 2001), pp. 741, 2001

[14] J. J. Bardyn et al., "Une architecture VLSI pour un operateur de filtrage median", Congeres reconnaissance des forms et intelligence artificielle, vol. 1, pp. 557-566, 1984

[15] Rafael C. Gonzalez, "Digital Image Processing", Pearson, 2009

[16] I. Sobel and G. Feldman, "A 3x3 Isotropic Gradient Operator for Image Processing", in R. Duda and P. Hart(Eds.), Pattern Classification and Scene Analysis, pp.271-272, 1973

[17] J. Canny, "A computation approach to edge detection", IEEE Transactions on Pattern Analysis and Machine Intelligence 8, pp.679-714, 1986

[18] Foley, van Dam, Feiner, and Hughes, Computer Graphics PRINSIPLES AND PRACTICE, Addison-Wesley Publishing Company, 1993

[19] McQueen, J. "Some methods for classification and analysis of multivariate observations", In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp.281-297, 1967

[20] S. P. Lloyd, "Least squares quantization in PCM", Special issue on quantization, IEEE Trans. Inform. Theory, 28, pp.129-137, 1982

[21] M. J. Swain and D. H. Ballard, "Color Indexing", International Journal of Computer Vision 7, pp.11-32, 1991

[22] B. Schiele and J. L. Crowley, "Object recognition using multidimensional receptive field histograms", European Conference on Computer Vision, vol. 1, pp.610-619, 1996