Automatic Text Imprint Analysis from Pill Images

Siroratt Suntronsuk
Department of Computer Science
Faculty of Science and Technology, Thammasat University
Pathum Thani, Thailand
siroratt_s@sci.tu.ac.th

Sukanya Ratanotayanon
Department of Computer Science
Faculty of Science and Technology, Thammasat University
Pathum Thani, Thailand
sratanot@cs.tu.ac.th

Abstract— **Pill identification is a serious concern for pharmacists due to similarity of pill appearances. Pill imprints usually contain important information that can be used to add or search for pill information on existing pill databases. However, current techniques for extracting imprints often give results as vectors which cannot be used with existing databases. Thus, this paper proposed an approach for automatically extracting text on imprints. This approach relied on a set of rules to locate imprint locations and a noise elimination technique to remove problematic pixels in binary images obtained by OTSU's thresholding. Our approach can achieve an F-measure of 0.77 for printed imprints and 0.57 overall.**

Keywords-Pill identification; text recognition; image processing; K-means clustering; Otsu's thresholding

I. INTRODUCTION

Because appearances of many pills are very similar, it is very difficult for pharmacists to identify unlabeled or unwrapped pills. This difficulty is one of an important concern as many situations arise for identifying unlabeled pills. For example, a patient may transfer to a new treatment facility but do not have his medical records transferred. There are existing online pill databases that can be used to search for pills. Most pill databases are maintained by foreign organizations and are focusing on updating pills used in their countries. Therefore, they often lack information of pills manufactured in Thailand.

Current researches aim to find a method for automatically extracting pill information from pill images. The information can be used to support automatic pill identification such as by supporting automatic update of pill databases, or performing pill identification with image search.

Pill imprints usually contain important information such as brands, and doses. It would be beneficial to be able to automatically extract texts on the imprints. The results can be extended to use for adding or searching pill information on existing pill databases using pill images. There are existing works [1, 2] that aim to detect pill imprints. However, these works extract imprints as vectors of feature descriptors which can't be used with existing pill databases that store imprint information as text.

This paper proposed an approach for pill imprint extraction which results in texts of the imprints. The proposed method begins by normalizing a pill image to a specific size so that rules for text area detection presented in our previous work [4] can be applied. Once the text area of the imprint is identified, the area is cropped and binarized. This paper compared two main binarization methods: Otsu's thresholding with noise elimination and K-means clustering [5] methods. Lastly, the binarized image is inputted to Tesseract that is trained with fonts from a training set.

An experiment was performed to evaluate the proposed approach using 540 pill images from Drug.com [6]. Among them, 180 images contained printed imprints and 360 images contained engraved imprints. The approach was implemented in Python with OpenCV framework [7]. The results showed that Otsu's thresholding with noise elimination method outperformed K-means clustering method for this task. The best result which was obtained by binarizing the imprint's area with Otsu's thresholding with noise elimination and using trained Tesseract [8], can reach the precision and recall over 57%.

II. RELATED WORK

Two common approaches for pill identification are i) online pill databases that a user can manually input appearances of a pill to search for its information, and ii) systems that can be searched by using pill images.

Many web-based databases are available to allow a user to manually searches for a pill by inputting its characteristics such shapes and colors via an online form. Example of these databases are Pillbox [9], MIMS Thailand [10], Lexicomp [11], epocrates [12], YaAndYou [13], WebMD [14], RxList [15], Healthline [16], DailyMed [17], and Drug.com [6]. These databases store searchable pill information including imprints information as text.

However, as most of available databases are maintained by foreign organizations, they lack information of medicines that are local made in Thailand. MIMS Thailand [10] does contain some Thai medicines made by big companies, but information of many local made pills are still not available. Thai food and drug administration (Thai FDA) database [18] and YaAndYou contain information of all pills that are allowed to be distributed by Thai FDA. However, both do not store pill appearance information. Therefore, they are of little use as a pill identification tool.

Other works aim to allow users to directly identify a pill using only its image. These approaches would automatically extract pill outward characteristics from its image instead of

978-1-4673-9077-4/17/$31.00 ©2017 IEEE

having a user keys in the information. The extracted information is usually in the form of feature descriptor vectors instead of text.

Chen [2],[19] proposed an imprint descriptor named Two-step Sampling Distance Sets (TSDS). This description was represented as a vector that contains information about shape and color of a pill. Caban [3] used a modified shape distribution technique to extract pill imprint as a vector descriptor. Annasaro [20] proposed a system for identifying a pill based on its shape, color and text imprint. In this system, a vector of an imprint descriptor is extracted using an Alphabet distance set. Although these existing methods used imprints as one of their features for identifying a pill, the information was represented as vectors. Therefore, the extracted information cannot be used with the existing pill databases.

For techniques that extract information of a pill from its picture, there are common difficulties arising from conditions of the image such as a poor image resolution, over/under exposure, unwanted shadows, and poor/high contrast. There are some works that aim to deal with these difficulties. Mancas [21] proposed a contrast enhancement method by imitating human retina behavior. Wang [22] suggested a multi-view based method to remove specular reflections from images of pills inside transparent plastic packages. Chen [2] proposed dynamic contrast adjustment to solve an exposure problem using the V channel from the HSV color system.

III. PILL IMAGE PREPARATION PROCESS

As images of pill can have different resolutions, we reduced the effect by normalizing the size of all images. First, we located the area of a pill in an image and crop it. To do this, we applied K-means clustering on the image with K=3 as most pills only have 2 different dominant colors. The cluster with the most pixels on the border were considered the background. All pixels in the background cluster were eliminated. Then we applied Canny edge detection to find edges of the pill. Using the edge map, we located a bounding box that covered the largest contour which was considered the pill. Then we cropped the image using the bounding box area. Once the image was cropped, we used Bicubic interpolation [23] to make the images had 300 pixels in height while maintaining the same aspect ratio as the original image.

Most engraved imprints usually have a weak contrast between the background and the imprint. This is because the color differences between the imprint area and background are only caused by shadows casting into the engraved areas. Therefore, the image needed to be enhanced to obtain clearer edges.

Gastal [24] proposed an edge-aware smoothing technique that can eliminate noise pixels whilst retains edge contrast. This method performs equally well compared to popular filters such as Anisotropic diffusion [25] and Bilateral filter [26], but uses less execution time. The results from this filter will also have less noise pixels comparing to the Histogram equalization method.

Therefore, to improve imprint contrast, we applied Gastal's detail enhancing filter [24] with $\sigma s = 10$ and $\sigma r = 0.15$ on pill pictures. As contrast increases, so are noise pixels. To reduce noises, we used Edge preserving filter [24] with $\sigma s = 35$ and $\sigma r = 0.4$. This filter uses less computational time compared to the standard smoothing filter. In addition, it also maintains contrast of edges while smoothing noise pixels. Examples of original images and results after preprocessing are shown in Figure 1 and Figure 2 respectively.

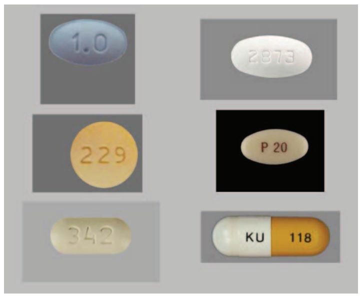

Figure 1. Original images

Figure 2. Images after preparation process

IV. TUNING PROCESS

Two types of tuning were done to obtain our results: filter parameter tuning and Tesseract font training. For both trainings, we used a dataset of 200 images from U.S. National Library of Medicine (NLM) system.

TABLE I. TEXT BOX DETECTION RESULTS

	Original paper	Tuned parameters
Imprint area detection accuracy (%)	79.44	89.814

Gastal's filters used for preprocessing need two parameters σs (Size of neighborhood for smoothing) and σr (Allowed levels of differences between neighbor pixels). We applied our techniques on the training set. To tune the parameters, when applying Gastal's filters, we varied parameters σs and σr. The σs was varied from 0 to 200, and the σr was varied from 0 to 1. The results were evaluated for the accuracy of the text box area detection. The value of $\sigma s = 10$ and $\sigma r = 0.15$ gave the best results. Using the new parameter values, the accuracy of the text box detection increased about 10% from using the parameter in our previous study [4]. The result is presented in Table I.

We use Tesseract for recognizing characters of the imprints. Our compared methods for binarization, Otsu thresholding and K-means clustering, output characters with very different appearances (very thin characters for Otsu thresholding and very thick characters for K-means clustering.) Therefore, we needed to train Tesseract to recognize fonts that are similar to each type of characters. The training was done with images of characters obtained from the binarization step of each approach. The images were slightly modified to imitate fonts in various situations.

Training a new font in Tesseract involves 3 steps: a) preparing the character image files, b) creating character labels, and c) creating Tesseract training files.

A. *Character image file preparation*

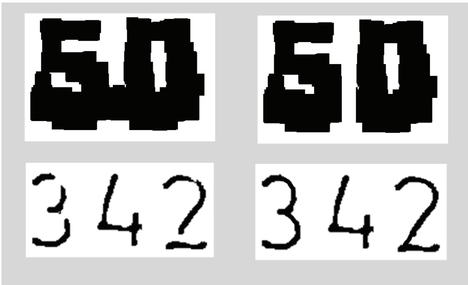

Figure 3. modified character images

This step created a .tiff image file that contained images of characters of the new font. We collected images of characters from binarized images of the training set. Some character images were slightly modified to make them more complete, but still resemble the original appearance of the font. Example modification is shown in Figure 3. We could collect images of all number (0-9), but could not collect images of some characters as they do not appear in the training set. We used 15 different character images for each number. Once all images were prepared, we created a .tiff image file that contained all character images with the resolution 300x300 for training.

B. *Character Label Creation*

This step created a .box file that contains labeling information for each character such as its position in the tiff image, and a tag identifying the character. The basic .box file was created with bbTesseract provided with Tesseract. Then, we used jTessBoxEditor, an editor with a Graphical User Interface, to correct information of each training character. Example labeling information of characters were shown in Figure 4.

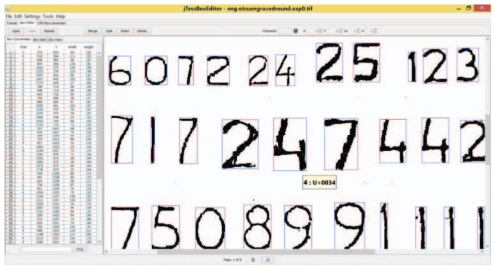

Figure 4. Exsample label information in jTessBoxEditor

C. *Tesseract Training File Creation*

Using the .box and .tiff files created in the previous steps, we created 8 data files required for generating a traineddata file needed for training a new font.

Files 'inttemp', 'pffmtable' and 'normproto' contains character shape feature information. The 'unicharset' file contains character properties, e.g. whether a character is a digit, an uppercase character or a lowercase character.

Tesseract uses three dictionary files for each new font. We disabled the use of 'freq-dawg' and 'word-dawg' that store frequent words and possible words respectively. These dictionaries are not suitable for our application as imprints are usually not common words. We created a custom dictionary using 'user-words' instead. The file was populated with all imprints available in drug.com at the current time of this study.

DangAmbigs contains ambiguous characters or set of characters which can be manually specified to Tesseract. However, we did not use this formation in the training process.

In addition to training new fonts, we also needed to configure Tesseract properties to be suitable to use with imprints. Imprints often contain abbreviation such as an abbreviation of a manufacturer's name. Therefore, we have to disable the use of the system dictionary and the default English dictionary language to prevent from Tesseract misinterpreting the abbreviation with common words.

V. PILL IMAGE RECOGNITION PROCESS

To recognize text on imprints, each pill images was processed with the following 3 steps. First, we located the imprint area so that we could crop them from the image. This step was done to eliminate unrelated pixels that could be noises. The cropped image is converted to a binary image and then passed along to Tesseract to identify characters. The details of these steps are described below.

A. *Finding imprint area*

The goal of this step is to locate a smallest bounding box that contains all text in an imprint. We called this box a "text box". We used the same method presented in [4] to determine the area of the text box. This method first created an edge map for each R (E_R), G (E_G) and B (E_B) channel of the image. Then a combined edge map (E) was created by combining all edges in the previous three maps.

$$E = E_R \cup E_G \cup E_B$$

Once a combined edge map was created, dilation and closing morphology were applied on the edge map to enlarge edge pixels and reduce holes inside imprint characters, respectively. Then, based on the specified rules, bounding boxes of characters were identified and combined to create the text box. Finally, we cropped the image using the text box so that the image only contained the actual area of the imprint.

B. Image binarization

Once the actual imprint area was identified, the imprint was binarized to get a black and white image that can be used with Tesseract. Binarization step was very important because noises in the images could cause Tesseract to not recognize characters. In this paper, we compared two binarization approaches. The first approach is based on Otsu global thresholding [23] used in our previous paper. The second method is based on K-means clustering used in [5] to binarize an image of characters etched on metal surfaces. Details of each binarization approach are as follows.

1) Otsu global thresholding

Otsu global thresholding method is a greedy method that divides pixels into two classes by searching for a threshold color value that minimizes the variance within a class while maximizes the variance between two classes. For this binarization method, the image was first converted to grayscale for further processing.

In our preliminary study with the training set, we found that there were usually unwanted noises caused by shadows of engraved imprints or multi-color noises for printed imprints. Therefore, we added a process to eliminate these noises from the binary image. The comparison of the binarization process with and without noise elimination is shown in Figure 5.

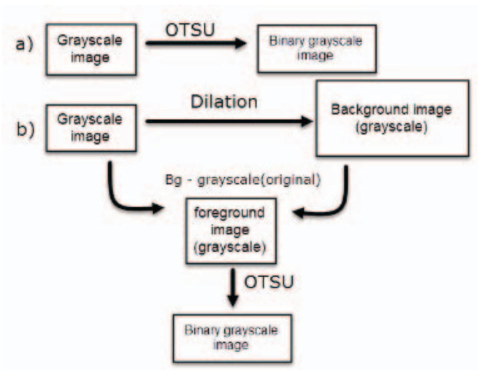

Figure 5. The flowchart of a) Original Otsu thresholding b) Otsu thresholding with noise elimination

The first step of noise elimination is construction of a background image using dilation morphology with circle kernel size 6*6 pixels. An example result of a background image is shown in Figure 6b.

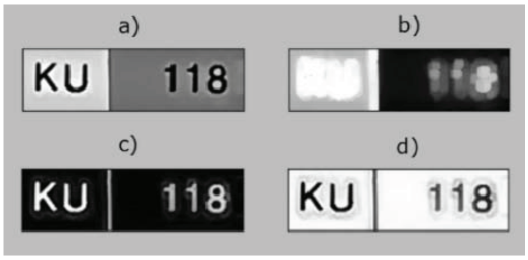

Figure 6. a) grayscale image of an imprint area b) background image c) foreground image d) reverse color of c)

Then, we subtracted the background image with the original grayscale imprint image to get rid of background pixels. The result was a foreground image with black background. Lastly, the foreground image was complemented to reverse the color so that we had a black imprint on a white background. Once noises in the background were eliminated, we performed Otsu global thresholding to convert a foreground image to a binary image. The results of each steps were shown in Figure 6, and example binarized results were shown in Figure 7.

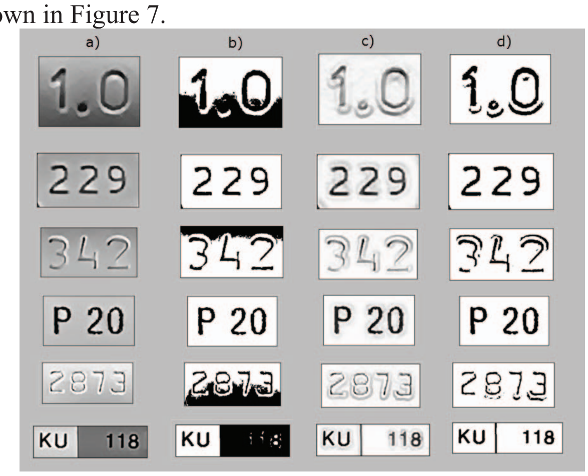

Figure 7. Column a) grayscale image of an imprint area b) binary image from Otsu c) grayscale image from Noise elimination d) Binary image from Otsu's thresholding with Noise elimination

2) K-means clustering binarization

K-means clustering is an unsupervised learning algorithm for categorizing a set of data into K clusters. This algorithm starts by randomly initiating K centroids for each cluster. Then each data is assigned to a cluster with the centroid that has the closest distance to the data. When all data are assigned, the centroid of each cluster is recalculated. The process of assigning data and recalculating the centroids repeats until no centroid is changed in the recalculation step.

We applied K-means clustering with K value equals to 3. The three clusters were; foreground (imprint) cluster, background cluster and noise cluster. To overcome blurring and lighting effects, we applied contrast enhancement ($I_{enhanced}$) [21] for each of the RGB channel before performing K-means clustering.

We determined the type of each cluster using the following rules. The cluster that contained the most pixels on the images border were considered the background cluster. For the remaining two clusters, one would be the foreground cluster. Since the foreground cluster contained text, therefore the amount of each connected components in the cluster should be similar. We calculated the standard deviation (SD) of number of pixels in connected components of both clusters. The cluster that had higher SD was considered a noise cluster.

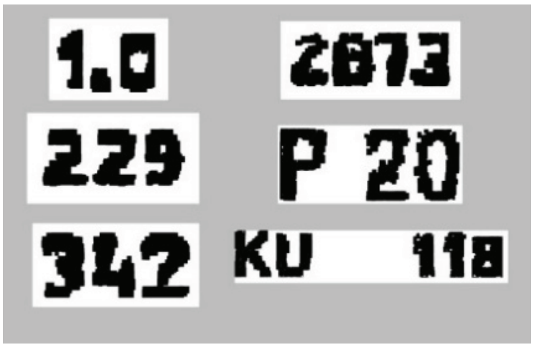

Figure 8. Binary images from K-means clustering

C. Imprint recognition

Once an imprint image was binarized, we inputted the binary image to Tesseract [8], which is a well-known optical character recognition (OCR) engine, to obtain the texts on the imprint. In this study, we used Tesseract version 3.05. We trained Tesseract to recognize the fonts from both binarization approaches using the method presented in section IV. A total of five fonts were trained. For each binarization approach, we trained separate fonts for printed and engraved imprint characters. For Otsu thresholding, we trained an additional font for a set of very slim characters.

VI. EVALUATION

We evaluated the proposed method using 540 pill images randomly selected from an online pill database “drug.com”. The selected images included ones with most common pill shapes (round, oval and capsule) and colors (white, yellow, orange and blue) [3]. Among 540 images, 360 images had engraved imprints and 180 images had printed imprints. Printed imprint could be both single color and multi-color. Also, in this study we selected only pills that do not contain special symbols in their imprints. The proposed approaches were implemented with OpenCV 3.0 library and Python 2.7.

We used precision, recall and f-measure as indicators to measure performance of the approaches. As the proposed approaches outputted a string of a pill imprint, we defined a set of relevant and retrieved items that will be used to calculate these indicators as follows:

- Each character in the Tesseract output of each imprint was considered as a retrieved item.
- Each correct character of the text imprint was considered as a relevant item.

The precision, recall and f-measure are defined as follows:

$$precision = \frac{|\{relevant\ items\} \cap \{retrieved\ items\}|}{|\{retrived\ items\}|}$$

$$recall = \frac{|\{relevant\ items\} \cap \{retrieved\ items\}|}{|\{relevant\ items\}|}$$

$$F - Measure = \frac{2 \times precision \times recall}{precision + recall}$$

Table II showed the average precision, recall and f-measure of character recognition for all compared methods. The compared methods were divided into two main groups which were methods based on Otsu thresholding and K-means clustering. For images that were binarized with Otsu thresholding, we also compared the results between i) using untrained parameters for Gastal’s filters (same parameter as in [4]) and ii) Using trained parameters. We found that the trained parameters slightly increase the detection accuracy. When combined the trained parameter with noise elimination and Tesseract Training, we could increase the F-measure by 12%.

The surprising result was that K-means clustering method performed much worse than the Otsu thresholding, even after training Tesseract. This might be because the images received from this binarization results in large characters. These characters sometimes touched each other which made it difficult for Tesseract to identify each individual character.

Table III and Table IV elaborated on the Otsu thresholding results in Table II by differentiating between results from engraved (Table III) and printed (Table IV) imprints. Both methods performed better with printed imprints than with engraved imprints. This is because printed imprints have more distinct color between imprints and background. This led to clearer edges. For printed imprints, Otsu thresholding, with noise elimination and trained Tesseract could achieve F-measure at 77.3%. We found that, for printed imprints, noise elimination helped improve the result by around 10%. The cause of the improvement was that noise elimination can clean up binarization noises in pills that have multi colors such as capsules.

VII. CONCLUSION

This paper proposed a method for extracting pill imprint as text. The proposed method composed of 5 steps. First, images were normalized and enhanced to improve imprint’s contrast. Next, the imprint area was identified using 5 specified criteria. Then, we cropped and binarized the imprint area with i) Otsu’s thresholding with Noise elimination or ii) K-means clustering. Finally, the binarized result was inputted to a trained Tesseract to obtain text. The evaluation results showed that Otsu’s thresholding outperformed K-means clustering method for both engraved and printed imprints. Using Otsu’s thresholding binarization with noise elimination and trained Tesseract can achieve precision and recall over 57%. The proposed method still need improvement with binarizing engraved imprints.

TABLE II. AVERAGE PRECISION, RECALL AND F-MEASURE OF OCR RESULTS OF EACH APPROACH

	Untrainted parameters & Otsu Thresholding	Trained parameter & Otsu Thresholding				K-means clustering	
		Without Noise elimination		With Noise elimination			
		Without Trained OCR	*Trained OCR*	*Without Trained OCR*	*Trained OCR*	*Without Trained OCR*	*Trained OCR*
Precision	0.465	0.491	0.544	0.537	0.585	0.036	0.055
Recall	0.464	0.514	0.541	0.562	0.588	0.045	0.061
F-measure	0.467	0.502	0.542	0.548	0.586	0.04	0.058
Average execution time(Millisecond)	9.254	9.672	11.731	11.470	13.278	15.883	16.922

TABLE III. AVERAGE PRECISION, RECALL AND F-MEASURE OF OCR RESULTS OF OTSU

	Engraved imprints					Print Imprints				
	Original imprint area with OTSU[4]	Without Noise elimination		With Noise elimination		Original imprint area with OTSU[4]	Without Noise elimination		With Noise elimination	
		Without Trained OCR	*Trained OCR*	*Without Trained OCR*	*Trained OCR*		*Without Trained OCR*	*Trained OCR*	*Without Trained OCR*	*Trained OCR*
Precision	0.280	0.308	0.390	0.314	0.388	0.659	0.674	0.698	0.759	0.781
Recall	0.304	0.345	0.399	0.360	0.410	0.624	0.683	0.683	0.764	0.765
F-measure	0.292	0.325	0.394	0.335	0.399	0.641	0.678	0.690	0.761	0.773

TABLE IV. AVERAGE PRECISION, RECALL AND F-MEASURE OF OCR RESULTS OF K-MEANS CLUSTERING

	Engraved imprints		Print Imprints	
	Without Trained OCR	*Trained OCR*	*Without Trained OCR*	*Trained OCR*
Precision	0.018	0.035	0.053	0.075
Recall	0.024	0.039	0.066	0.082
F-measure	0.021	0.037	0.059	0.078

REFERENCES

[1] T Kasar, J Kumar, and A G Ramakrishnan, “Font and background color independent text binarization,” 2007.
[2] J. Yu, Z. Chen, and S. Kamata, “Pill Recognition Using Imprint Information by Two-Step Sampling Distance Sets,” 2014, pp. 3156–3161.
[3] J. J. Caban, A. Rosebrock, and T. S. Yoo, “Automatic identification of prescription drugs using shape distribution models,” 2012, pp. 1005–1008.
[4] S. Suntronsuk and S. Ratanotayanon, “Pill Image Binarization for Detecting Text Imprints,” presented at the International Joint Conference on Computer Science and Software Engineering, Thailand, 2016, vol. 13.
[5] C. Mancas-Thillou and M. Mancas, “Comparison between Pen-scanner and Digital Camera Acquisition for Engraved Character Recognition,” presented at the International workshop on camera-based document analysis and recognition (CBDAR 2007), Curitiba, Brazil, 2007, pp. 130–137.
[6] “Drugs.com Pill Identifier.” [Online]. Available: http://www.drugs.com/pill_identification.html. [Accessed: 07-Nov-2016].
[7] “OpenCV 3.1.0.” [Online]. Available: http://code.opencv.org/projects/opencv. [Accessed: 07-Nov-2016].
[8] R. Smith, “An Overview of the Tesseract OCR Engine,” 2007, pp. 629–633.
[9] “‘Pillbox’ National Library of Medicine, National Institutes of Health, United States.” [Online]. Available: http://pillbox.nlm.nih.gov/. [Accessed: 07-Nov-2016].
[10] “Image Advanced search-MIMS thailand.” [Online]. Available: https://www.mims.com/Thailand/image/AdvancedSearch. [Accessed: 07-Nov-2016].
[11] “Lexicomp - drug information.” [Online]. Available: https://itunes.apple.com/th/app/lexicomp/id313401238?mt=8. [Accessed: 07-Nov-2016].
[12] “Epocrates-an Athenahealth service.” [Online]. Available: http://www.epocrates.com/. [Accessed: 07-Nov-2016].
[13] “YaAndYou.” [Online]. Available: https://itunes.apple.com/th/app/yaandyou/id459400481?mt=8. [Accessed: 07-Nov-2016].
[14] “WebMD pill identification tool.” [Online]. Available: http://www.webmd.com/pill-identification/. [Accessed: 07-Nov-2016].
[15] “RxList pill identification tool.” [Online]. Available: http://www.rxlist.com/pill-identification-tool/article.htm. [Accessed: 07-Nov-2016].
[16] “Healthline pill identifer.” [Online]. Available: http://www.healthline.com/pill-identifier. [Accessed: 07-Nov-2016].
[17] “DAILYMED-National library of medicine.” [Online]. Available: http://dailymed.nlm.nih.gov/dailymed/index.cfm. [Accessed: 07-Nov-2016].
[18] “Pill information service, Thai food and drug administration.” [Online]. Available: http://fdaolap.fda.moph.go.th/logistics/drgdrug/Dserch.asp. [Accessed: 07-Nov-2016].
[19] Z. Chen, J. Yu, S. Kamata, and J. Yang, “Accurate system for automatic pill recognition using imprint information,” IET Image Process., vol. 9, no. 12, pp. 1039–1047, Dec. 2015.
[20] E.Annasaro et al, “Color and Shape Feature Extraction and Matching in Pill Identification Systems,” IJCSIT, vol. 5, no. 2, pp. 1011–1015, 2014.
[21] M. Mancas, C. Mancas-Thillou, B. Gosselin, and B. Macq, “A Rarity-Based Visual Attention Map - Application to Texture Description,” 2006, pp. 445–448.
[22] C. Wang, S. Kamata, and L. Ma, “A fast multi-view based specular removal approach for pill extraction,” 2013, pp. 4126–4130.
[23] R. C. Gonzalez and R. E. Woods, Digital image processing, 3rd ed. Upper Saddle River, N.J: Prentice Hall, 2008.
[24] E. S. L. Gastal and M. M. Oliveira, “Domain transform for edge-aware image and video processing,” ACM Trans. Graph., vol. 30, no. 4, p. 1, Jul. 2011.
[25] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 7, pp. 629–639, Jul. 1990.
[26] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” 1998, pp. 839–846