J Electr Eng Technol.2015;10(1): 401-407
http://dx.doi.org/10.5370/JEET.2015.10.1.401

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Improved Spam Filter via Handling of Text Embedded Image E-mail

Seongwook Youn* and Hyun-chong Cho†

Abstract – The increase of image spam, a kind of spam in which the text message is embedded into attached image to defeat spam filtering technique, is a major problem of the current e-mail system. For nearly a decade, content based filtering using text classification or machine learning has been a major trend of anti-spam filtering system. Recently, spammers try to defeat anti-spam filter by many techniques. Text embedding into attached image is one of them. We proposed an ontology spam filters. However, the proposed system handles only text e-mail and the percentage of attached images is increasing sharply. The contribution of the paper is that we add image e-mail handling capability into the anti-spam filtering system keeping the advantages of the previous text based spam e-mail filtering system. Also, the proposed system gives a low false negative value, which means that user's valuable e-mail is rarely regarded as a spam e-mail.

Keywords: E-mail classification, OCR, Ontology, Spam filtering

1. Introduction

With the advances in technology, these sensors are becoming less expensive and smaller, making them widely and easily available for commercial use and also extensively for research as well. Since the opening of the internet in early 1990's, the continuous growth of spam phenomena has become a major problem for corporate, private users, internet service providers. Spam causes e-mail systems to experience overloads in bandwidth and server storage capacity, with an increase in annual cost for corporations of over tens of billions of dollars.

Anti-spam is a very active area of research, and various forms of filters, such as white-lists, black-lists, and content-based lists are widely used to defend against spam [1, 2]. Spam detection can be converted into text classification problem; many content-based filters utilize machine learning algorithms for filtering spam. The first countermeasures taken by spammers consisted in adding bogus text to their e-mails, usually taken from books or news articles, to compromise the effectiveness of statistical techniques. However, a new kind of trick introduced some years ago has rapidly spread during the past year and is now adopted in a large fraction of spam e-mails: it consists in embedding the spam message into attached images to circumvent all spam detection techniques based on the analysis of body text. This kind of spam is known as image spam. A number of spammers have been evading filters recently by encoding their messages as images and including some irrelevant good words. This implies the contents are hard to retrieve from the binary image encoding. This type of image spam accounts for 40% of all global spam in 2007, compared with just 1% in late 2005 [3].

By sending e-mails that contain no text, only pictures, or along with irrelevant good words, spammers have found that they can evade many security systems. The messages often include image files that have a screen shot offering the same types of information advertised in traditional text-based spam.

In a citation, among around 21,000 spam e-mails collected by [4] in their personal mailboxes from October 2004 to August 2005, 4% contained attached images. The percentage of attached images is increased to 25% in spam e-mails collected between April and August 2006. Among 143,061 spam e-mails donated by end users throughout 2005 to the 'submit' archive of the publicly available Spam Archive corpus, 9% contained attached images, while the percentage increased to 17% among the 18,928 spam e-mails posted between January and June 2006 [4, 5]. This implies that the percentage of spam e-mails incorrectly labeled as legitimate by current spam filters can increase significantly.

According to IBM X-Force's "Mid-Year Trend and Risk Report" [6], the ratio of image spam had been declined to under 1% in the end of 2007, and continued to be very low until the middle of 2011. There was the rebirth of image spam from 2011 to the mid of 2012 (around 8%), but after that it had disappeared again. However, we still have to handle image spam. Accordingly, improving content-based spam filters with the capability of analyzing text embedded into attached images is becoming a relevant issue given the current spam trend.

The main contributions of the previous system [7] is to create a spam filter in the form of ontology, which is user-customized, scalable, and modularized, so that it can be embedded to many other systems for better performance.

† Corresponding Author: Dept. of Electrical and Electronics Engineering, Kangwon National University, Chuncheon, 200-701 Korea. (hyuncho@kangwon.ac.kr)
* Dept. of Computer Science, University of Southern California, CA, 90089-0781, USA. (syoun@usc.edu)
Received: March 14, 2014; Accepted: September 1, 2014

However, the previous system handles only text e-mail and the percentage of attached images is increasing sharply. The contribution of the paper is that we add image e-mail handling capability into the previous anti-spam filtering system keeping the advantages of the previous text based spam e-mail filtering system.

The rest of this paper is organized as follows. Section 2 reviews related work on image spam e-mail filtering methods. Section 3 describes the method to retrieve text from text embedded images using OCR [4]. Section 4 details the spam filtering system we developed. Section 5 presents experimental results. Section 6 compares the proposed system with commercial filters and finally section 7 concludes this paper.

2. Related Work

Gupta et al. proposed a way to overcome to certain limitations due to embedded obfuscation like complex backgrounds, compression artifacts and wide variety of fonts and formats. Their methodology consists of 4 steps (Identification of noise; Extraction of low level features or Calculation of entropy; Removal of noise; Content extraction using OCR) [8]. Their method showed about an accuracy of 93.3%. Woods et al. tried to show that using the low level image feature - edge, as well as the magnitude of the edges per image, it is possible to analyze and classify an image as spam or ham. They employed the Sobel edge detection algorithm, which analyzes a low level feature of an image as an alternative to the OCR only based filtering system [9]. Dredze et al. tried to automatically classify an image as being spam or legitimate e-mail. They presented features that focus on simple properties of the image, making classification as fast as possible. Their evaluation showed that they accurately classify spam images in excess of 90% and up to 99% on real world data [10].

Attempts to use OCR (Optical Character Recognition) techniques to convert spam images back to text for processing by text-based filters have been foiled. The goal of OCR is to classify optical patterns corresponding to alphanumeric or other characters. The process of OCR involves several steps including segmentation, feature extraction, and classification. An effective response by spammers is the application of CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) [11] techniques, which are designed to preserve readability by humans but capable of effectively confusing the OCR algorithms [12].

Sahami et al. [13], Graham et al. [14], and Zhang et al. [15] investigated the use of text categorization techniques based on the machine learning and pattern recognition approaches for e-mail semantic content analysis. With respect to manually encoded rules, using these techniques, categorization rules are automatically created and the system is generalized potentially.

One of the most popular anti-spam solutions is Spam Assasin [16] and there are several sites hosting plug-in rule modules. The SpamAssasin was created to make a general purpose system compatible with a variety of anti-spam filters. Towards this end, they created a new binary prediction problem: Is this image spam or legitimate e-mail? The classification can then be fed into existing content filters as a feature. Others have followed this approach [17] and it had several advantages. First, it separates image classification from spam e-mail classification, which is a difficult and well-studied problem. Second, e-mails can contain multiple images and it is not clear how to combine them towards a single prediction. One approach has been done by Aradhye et al. [17]. They treated each image separately, avoiding this difficulty. Finally, they did not commit to a specific content filtering system. Rather, they provided a single feature that can be integrated with any learning based anti-spam system.

Al-Duwairi et al. proposed image texture analysis based spam image filtering technique using low-level image features (color, shape, texture, etc.) detection for image characterization [18].

Attar et al. surveyed and explained many image spamming techniques, anti-image spamming techniques. Also, they discussed how to cope with those spamming techniques [19].

Spammers are embedding the e-mail's message into images sent as attachments, which are displayed by most e-mail clients. This can make all content based filtering techniques based on the analysis of plain text in the subject and body fields of e-mails ineffective. This trick is often used in phishing e-mails, which are one of the harmful spam e-mails. Among commercial and open source spam filters currently available, only a plug-in of the Spam-Assasin spam filter can analyze text embedded images, but it provides only a Boolean attribute indicating whether more than one word among a given set is detected in the text extracted by an OCR from attached images. Texts extracted through the OCR are used as training data set in the text based spam e-mail filtering system.

3. Retrieval of Text from Text Embedded Images using OCR

OCR translates images of text, such as scanned documents, into actual text characters. Also known as text recognition, OCR makes it possible to edit and reuse the text that is normally locked inside scanned images. OCR works using a form of artificial intelligence known as pattern recognition to identify individual text characters on a page, including punctuation marks, spaces, and ends of lines. OCR SDK was an important component in our project because its efficiency can critically affect our project. For this we initially selected three OCR Software

Development Kits, JOCR [20], Simple OCR [21] and Asprise OCR [22].

By running a sample of 200 image e-mails on these software's we determined that Asprise OCR was performing with an accuracy of 95%. It had the best detection rate among the three softwares and hence we decided to go with Asprise OCR for this project. The components of Asprise OCR for Java Asprise OCR comprise two essential components: A native library: AspriseOCR.dll [on Windows] and one Java package com.asprise.util.ocr [Cross platforms] main package; contains essential classes to perform OCR.

But the use of OCR tool is not cost effective with the large amounts of e-mails been handled daily by server-side filters. To address the issue, we suggest that computational complexity could be reduced by using a hierarchical architecture for the spam filter. Text extraction using OCR tool should be carried only if the previous less complex modules were not able to reliably detect whether an e-mail is legitimate or not.

As a training image data set, we prepared 1000 e-mails (819 image spam + 181 legitimate images). The Asprise OCR cannot handle 676 image e-mails because of image obscuring techniques like wave, animate, deform, rotate, etc. 12 image e-mails out of 143 image spam and 4 image e-mails out of 181 legitimate image e-mails are additionally missed by error. Finally, 131 text messages out of 143 and 177 text messages out of 181 are retrieved correctly. In the experiment, we used only image e-mails not using obscuring techniques. Fig. 1 is a snapshot of the implemented program.

- Source folder: Browse and select all image and text files
- Destination folder: All retrieved text through OCR and text files goes to destination folder
- File (for the third Browse tab): Specify a feature selection file containing feature list
- File (for the fourth Browse tab): Specify a path where the .arff (WEKA [23] input) file would be generated.

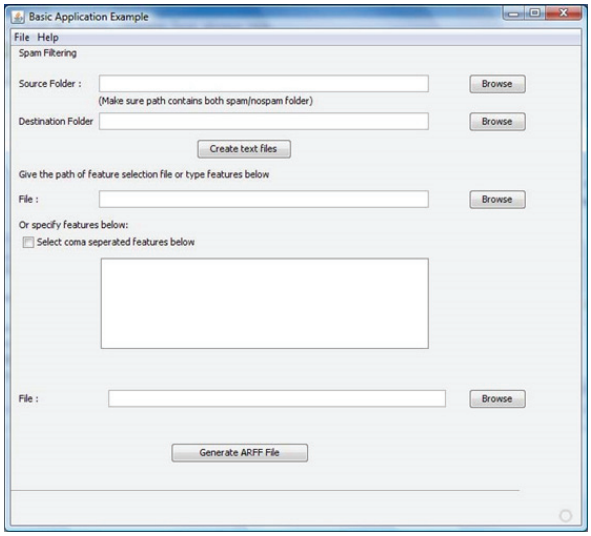

Fig. 1. Snapshot of OCR Implementation

4. Spam Filtering

Fig. 2 shows SPONGY(SPam ONtoloGY) framework to filter spam. The training data set is the set of e-mail that gives us a classification result. It is composed of both text e-mail and image e-mail. The test data is actually the e-mail will run through our system which we test to see if classified correctly as spam or not. This will be an ongoing test process and so, the test data is not finite because of the learning procedure.

Image e-mail among the training data set is entered into OCR, and then text information is retrieved from text embedded image e-mail. The training dataset was used as input to C4.5 classification. To do that, the training dataset should be modified as a compatible to query the test e-mail in Jena, an ontology should be created based on the classification result. To create ontology, an ontology language was required. RDF [24] was used to create an ontology. The classification result in the form of RDF file format was inputted to Jena, and inputted RDF was deployed through Jena, finally, an ontology was created.

Ontology generated in the form of RDF data model is the base on which the incoming mail is checked for its legitimacy. Depending upon the assertions that we can conclude from the outputs of Jena [25], the e-mail can be defined as spam or otherwise. The e-mail is actually the e-mail in the format that Jena will take in (i.e. in a CSV format) and will run through the ontology that will result in spam or not spam. The input to the system mainly is the training dataset and then the test e-mail. The test e-mail is the first set of e-mails that the system will classify and

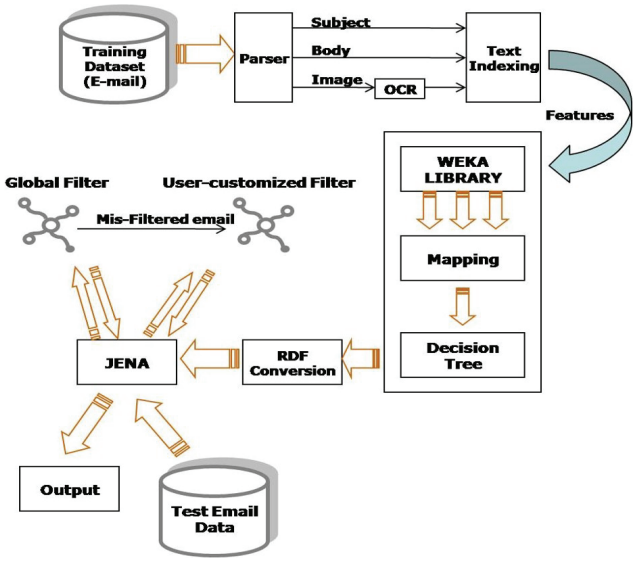

Fig. 2. SPONGY Architecture

learn and after a certain time, the system will take a variety of e-mails as input to be filtered as a spam or not. The training dataset which we used, which had classification values for features on the basis of which the decision tree will classify, will first be given to get the same. The classification results need to be converted to an ontology. The decision result which we obtained C4.5 classification was mapped into RDF file.

This was given as an input to Jena which then mapped the ontology for us. This ontology enabled us to decide the way different headers and the data inside the e-mail are linked based upon the word frequencies of each words or characters in the dataset. The mapping also enabled us to obtain assertions about the legitimacy and non-legitimacy of the e-mails. The next part was using this ontology to decide whether a new e-mail is a spam or not. This required querying of the obtained ontology which was again done through Jena. The output obtained after querying was the decision that the new e-mail is a spam or not.

Major trend in spam filtering area is a global filter. Generally, globally-trained filters outperform personally-trained filters for both small and large collections of users under a real environment. However, globally-trained filters sometimes ignore personal data. Globally-trained filters cannot retain personal preferences and contexts as to whether a feature should be treated as an indicator of legitimate e-mail or spam. Hence, we suggested two-level filters. Our goal is to combine the advantages of the both globally-trained filter and personally-trained filter for better spam filtering performance. That is the SPONGY (SPam ONtoloGY) system.

Spam e-mails vary from user to user and change over time, so learning and adaptive filtering is desirable. An ontology defines a common vocabulary to share information in a domain. It includes definitions of basic concepts in the domain and relations among them. Hence, an ontology could be developed to share common understanding of the structure of information among people or software agents, to increase reusability of domain knowledge, and to analyze domain knowledge. Several approaches adopted the machine learning techniques for learning and adaptation, but an ontology based filter is also proper for these necessities, so an ontology is used in our implementation. Ontologies allow for machine-understandable semantics of data, so by using an ontology as a filter, it can be embedded within other systems for better performance.

5. Experimental Results

In the experiment, we used both text e-mail and image e-mail. Data set was classified like the followings:

TS (Text Spam) -1008
TL (Text Legitimate) -1100
OCR IS (Retrieved text from Image Spam using OCR) – 131
OCR IL (Retrieved text from Image legitimate using OCR) – 177

We showed the experimental results in Table 1 to 4. We measured the false negative rate and false positive rate. The false negative rate is the proportion of positive instances that were erroneously reported as negative. It is equal to 1 minus the power of the test. The false positive rate is the proportion of negative instances that were erroneously reported as being positive. It is equal to 1 minus the specificity of the test. This is equivalent to saying the false positive rate is equal to the significance level.

False Negative rate =
# of false negatives / total # of positive instances
False Positive rate =
# of false positives / total # of negative instances

Actually, we used more Image data set, but many of them couldn't be handled by Asprise OCR. Hence, we considered only the image e-mails that can be handled by Asprise OCR. As you can see in the Tables 1, 2, 3 and 4, the SPONGY spam filter still showed good results without much performance degradation. Global filter in the Table is a spam filter excluding personalized filter in the SPONGY system. SPONGY is a spam filter created through the procedures in Fig. 2.

Table 1. Experimental results of global filter w/o OCR

	Without OCR functionality		
	Global filter (C4.5)		
	False negative	False positive	Correct classification
TS+TL	12.91%	3.67%	91.5085%

As a whole, in the SPONGY system with OCR functionality, false negative rate is increased from 6.34% to 7.36%, false positive rate is increased from 2.28% to 3.16%, and accuracy (Correct classification rate) is decreased a little. However, the SPONGY system got image e-mail handling capability.

6. Comparison with Commercial Filters

Spam e-mail filter is trying to block spam e-mail efficiently, but many spammers find new methods or techniques to try to break into the inbox of e-mail account of user. Most spam consists of an unwanted advertise, also some can transmit viruses, spyware on to your computer and cause problems. It is extremely annoying to go to user's inbox and have a look through a whole list of e-mail to find one legitimate e-mail. We did some experiment with Gmail, Yahoo! mail, the e-mail system of University of Southern California (USC). Also, we did a survey about

Table 2. Experimental results of SPONGY w/o OCR

	Without OCR functionality		
	SPONGY		
	False negative	False positive	Correct classification
TS+TL	6.34%	2.28%	95.4459%

Table 3. experimental results of global filter w/o OCR

	Without OCR functionality		
	Global filter (C4.5)		
	False negative	False positive	Correct classification
TS+TL+OCR IS+OCR IL	13.55%	4.74%	90.6043%

Table 4. Experimental results of SPONGY w/o OCR

	Without OCR functionality		
	SPONGY		
	False negative	False positive	Correct classification
TS+TL+OCR IS+OCR IL	7.36%	3.16%	94.6192%

some commercial spam filter programs.

6.1 Introduction to each e-mail system

Gmail has been known one of the best spam filters that prevent many spam e-mail to user's inbox. Spammers are finding it harder to send e-mail and evade the innovative spam block technology used by Gmail. In the Gmail, there is a "Report Spam" button. By clicking some message as "Report Spam", the Gmail will identify this type of message as spam the next time and not only block it at your e-mail account, everyone else's e-mail accounts will also block that message.

In case of image spam, it is difficult for Gmail to block, but with Optical Character Recognition (OCR), it can read what the image content says and block the messages. Gmail supports Sender Policy Framework (SPF), DomainKeys, which domain the message originates from, and DomainKeys Identified Mail (KIM), to verify the sender and help recognize whether it is real or forged messages. The sender cannot through the third party using multiple authentication system, which is different from many other webmail services support a single authentication system.

Yahoo! mail also supports many similar features used in Gmail. Yahoo! mail servers are going to need to separately check DomainKeys, SPF, and e-mail Caller ID. When user logs in his/her account, user can create user's own filter using filtering features provided by Yahoo! mail. User can specify whether or not the match should be case-sensitive and where the target string should appear in text that you are trying to match (ex. Contains, Does not contain, Begins with, and Ends with).

The USC e-mail system uses a centralized spam detection system from Symantec called Brightmail AntiSpam [26] that scans all incoming e-mail before the messages are delivered to the user's inbox. If the e-mail meets the specific criteria defined by the antispam filters, it tags the message header as potential spam. According to the announcement of the Brightmail AntiSpam, the false positive rate of the program is an extremely low 0.001%.

The SPONGY system uses two-level filter using dynamic ontologies: a first level global ontology filter and a second level user profile ontology filter. The user profile ontology filter is created based on the specific user's background as well as the filtering mechanism used in the global ontology filter creation.

6.2 Evaluation and comparison

We surveyed some commercial spam filters. We tested the filters using their 30 days trial versions.

Commercial spam filters supports many features as you can see in Table 5. Preset categories are provided by the program vendor freely. It contains content such as financial, adult content, health, etc. Rule customization option will allow you to add, remove, or modify the filtering rules. A rule is a set of criteria for determining whether or not an e-mail is spam or legitimate.

However, most of commercial filters are too complicated and difficult for the end users. As you can see, most of filters can allow or block IP address, server, and e-mail address. There are many other known filters in the world.

We also compared Gmail, Yahoo! mail, the USC e-mail and SPONGY. The detailed technology of commercial filters is not revealed, but most commercial filters are using data mining, previous personal e-mail data, other information which users did on the web, etc. SPONGY showed better false negative rate, which means that only a small portion of legitimate e-mail is classified as a spam e-mail. In the experiment, my e-mail addresses and messages are used. We cannot specify sender's e-mail address because most of e-mail systems support authentication system, hence I cannot test with other e-mail address. We did two experiments. The first experiment is performed with the same e-mail data set (e-mails used for the SPONGY system are forwarded to my account of Gmail, Yahoo! mail, and the USC e-mail).

Table 5. Spam filter review

	Spam Eater Pro	CA Anti-Spam	Choic Mail One	Spam Killer	SPONGY
Block IP addr	○		○		○
Block server	○		○	○	○
Block email addr	○	○	○	○	○
Blocklist support	○				○
Allow IP addr	○		○		○
Allow server	○		○	○	○
Allow email addr	○	○	○	○	○
Individual user profile	○		○	○	○
Repoting capabilities			○	○	○

E-mail addresses in hotmail.com and hanmail.net are used as a sender e-mail. In the second experiment, we just checked each e-mail system's filtering accuracy with false negative and false positive in my e-mail account of each e-mail system. The second experiment is done under more realistic environment. The experiment is done with own filters of each e-mail system (Gmail, Yahoo! mail, the USC e-mail, and SPONGY) with the default setting.

As you can see in Table 6, the experiment was performed with the same data set. E-mail data used in the SPONGY system experiment were sent to Gmail, Yahoo! mail, and the USC e-mail system. We used Hotmail and Hanmail as a sender e-mail account, and my e-mail accounts in Gmail, Yahoo! mail, and the USC e-mail system were used as a receiver. In the experiment, we cannot use other person's e-mail account because of privacy, and send bulk of e-mail because of authentication policy of each e-mail system.

Table 6. Comparison result with same e-mail dataset

	Google Gmail	Yahoo mail	USC mail	SPONGY
False negative	81.2705%	83.0357%	60.3603%	7.3610%
False positive	0.7463%	2.2388%	1.5152%	3.1607%

The SPONGY system is scalable, learning, multi-level filter. Although we consider experimental environment like the several difficulties, the SPONGY system showed better performance than Gmail, Yahoo! mail, and the USC e-mail system. In here, we insist that the experimental results of the SPONGY system in at least our experimental environments are efficient. False positive values of all the e-mail systems are reasonable, but false negative values of Gmail, Yahoo! mail, and the USC e-mail system are not good. Probably, it happened because the sender of e-mail is me, and most of the e-mail system considers the sender when their filtering policy is used.

Another experiment was performed on the real setting with different e-mail data set. In this case, the SPONGY system showed good performance in both false negative rate and false positive rate. In the SPONGY system, most balanced false negative and false positive rate values were obtained. False negative rate was 7.3610% and false positive rate was 3.1607%. Three other commercial mail systems showed low experimental results. False negative rate of the Yahoo! mail was extremely bad for us. Brightmail AntiSpam of the Symantec used in the USC e-mail system showed very low false positive rate. Generally, performance order is SPONGY, the USC e-mail system, Google Gmail, and Yahoo! mail. Experimental results are shown in Table 7. With the same e-mail data set, the SPONGY showed the best false negative rate. With some of test e-mail data set, SPONGY showed better performance at least under my experiment. By increasing image e-mail handling capability, we possibly increase the performance of the spam filtering system. We know our experiment is somewhat restricted, but it demonstrates the potential capability of the spam filtering system we proposed.

As you can see in Table 6 and 7, the proposed SPONGY system gives a low false negative value, which means that user's valuable e-mail is rarely regarded as a spam e-mail.

Table 7. Comparison result with different e-mail dataset

	Google Gmail	Yahoo mail	USC mail	SPONGY
False negative	11.2436%	19.3319%	9.6611%	7.3610%
False positive	4.6358%	6.3830%	2.8169%	3.1607%

5. Conclusion

We added image spam handling capability using OCR into the text-based anti-spam filtering system. By handling of text embedded image e-mail, the proposed system can be used partially for both text e-mail and image e-mail. The experiment was somewhat restricted, but it demonstrates the potential capability of the proposed system. However, to cope with the image e-mail thoroughly, we need to adopt advanced image processing techniques. Then, we can face image obscuring techniques like wave, animate, deform, and rotate. In the future, we will experiment with the combination of the general corpus data set and our data set for generality.

As we explained, new spamming technique appears continuously and traditional spamming technique is also prevailing. Spamming technique is evolutionary; hence the spam filtering technique must catch up with the new spamming technique.

Acknowledgements

This study was supported by 2014 Research Grant from Kangwon National University.

References

[1] H. Lam, D. Yeung, "A Learning Approach to Spam Detection based on Social Networks," In Proceedings of 4th Conference on E-mail and Anti-Spam, 2007.

[2] A. Pathak, S. Roy, Y. Hu, "A Case for a Spam-Aware Mail Server Architecture," In Proceedings of 4th Conference on E-mail and Anti-Spam, 2007.

[3] Spam Filter Review, 2007. http://spam-filter-review.toptenreviews.com.

[4] G. Fumera, I. Pillai, F. Roli, "Spam Filtering Based On The Analysis Of Text Information Embedded Into Images," Journal of Machine Learning Research, Volume 6, pp. 2699-2720, 2006.

[5] B. Biggio, G. Fumera, I. Pillai, F. Roli, "Image Spam Filtering Using Visual Information," In Proceedings

of ICIAP, pp. 105-110, 2007
[6] IBM X-Force Mid-Year Trend and Risk Report, "http://www-03.ibm.com/security/xforce/downloads.html"
[7] S. Youn, D. McLeod, "Spam E-mail Classification using an Adaptive Ontology," In Journal of Software, Volume 2, No. 3, pp. 43-55, Sep 2007.
[8] A. Gupta, C. Singhal, S. Aggarwal, "Identification of Image Spam by Using Low Level & Metadata Features," In International Journal of Network Security & ITS Applications, Volume 4, No. 2, Mar 2012.
[9] N. Woods, O. Longe, A. Roberts, "A Sobel Edge Detection Algorithm Based System for Analyzing and Classifying Image Based Spam," In Journal of Emerging Trends in Computing and Information Sciences, Volume 3, No 4, Apr 2012.
[10] M. Dredze, R. Gevaryahu, A. Elias-Bachrach, "Learning Fast Classifiers for Image Spam," In Proceedings of 4th Conference on E-mail and Anti-Spam, 2007.
[11] The CAPTCHA project, 2000. http://www.captcha.net.
[12] B. Byun, C. Lee, S. Webb, C. Pu, "A Discriminative Classifier Learning Approach to Image Modeling and Spam Image Identification," In Proceedings of 4th Conference on E-mail and Anti-Spam, 2007.
[13] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, "A Bayesian approach to filtering junk e-mail," In AAAI Technical Report WS-98-05, Madison, Wisconsin, 1998.
[14] P. Graham, "A plan for spam," http://paulgraham.com/spam.html.
[15] L. Zhang, J. Zhu, T. Yao, "An evaluation of statistical spam filtering techniques," In ACM Transactions on Asian Language Information Processing, Vol. 3, No. 4, pp. 243-269, 2004.
[16] The SpamAssassin project. http://spamassassin.apache.org/.
[17] H. Aradhye, G. Myers, J. Herson, "Image analysis for efficient categorization of image-based spam e-mail," In Proceedings of Int. Conf. Document Analysis and Recognition, pp. 914-918, 2005.
[18] Basheer Al-Duwairi, Ismail Khater, Omar Al-Jarrah, "Detecting Image Spam Using Texture Features", International Journal for Information Security Research (IJISR), Volume 2, Issues 3/4, pp. 344-353, September / December 2012
[19] Abdolrahman Attar, Reza Moradi Rad, Reza Ebrahimi Atani, "A survey of image spamming and filtering techniques," Artificial Intelligence Review, 40(1), pp. 71-105, 2013
[20] The JOCR. http://jocr.sourceforge.net/links.html.
[21] The SimpleOCR. http://www.simpleocr.com/.
[22] The Asprise OCR. http://asprise.com/product/ocr/selector.php.
[23] The WEKA. http://www.cs.waikato.ac.nz/ml/weka/.
[24] The RDF. http://www.w3.org/RDF/.
[25] The Jena. http://jena.sourceforge.net/.
[26] The The Brighmail AntiSpam by Symantec. http://www.symantec.com/business/products/overview.jsp?pcid=psc_msg_security&pvid=835_1.

Seongwook Youn He received the B.S. degree in Computer Science from Sogang University, Seoul, Korea in 1997, and M.S. and Ph.D. degrees in Computer Science from University of Southern California, Los Angeles, CA in 2002 and 2009, respectively. Dr. Youn's current interests are Spam e-mail filtering, Big Data Analysis, Recommendation system, etc.

Hyun-chong Cho He received the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Florida, USA in 2009. During 2010-2011, he was a Research Fellow at the University of Michigan at Ann Arbor, USA. From 2012 to 2013, he was a Chief Research Engineer in LG Electronics, South Korea. He is currently an Assistant Professor at Kangwon National University, South Korea.