AN INCREMENTAL LEARNING BASED FRAMEWORK FOR IMAGE SPAM FILTERING

Li Xiao Mang[1], HaRim Jung[1], Hee Yong Youn[1] and Ung-Mo Kim[1]

[1]School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea

ABSTRACT

Nowadays, an image spam is an unsolved problem because of two reasons. One is due to the diversity of spamming tricks. The other reason is due to the evolving nature of image spam. As new spam constantly emerging, filters' effectiveness drops over time. In this paper, we present an effective anti-spam approach to solve the two problems. First, a novel clustering filter is proposed. By exploring the density-based clustering algorithm, the proposed filter is robust to spamming tricks. Then, we present a hierarchical framework by combining the clustering filter with other machine learning based classifiers to further improve the filtering capacity. Moreover, incremental learning mechanism is integrated to ensure the proposed framework be capable of adjusting itself to overcome new image spamming tricks. We evaluate the proposed framework on two public spam corpora. The experiment results show that the proposed framework achieves high precision along with low false positive rate.

KEYWORDS

Image spam, anti-spam filter, framework, clustering, incremental learning

1. INTRODUCTION

Since 1990s, the problem of spam has become a serious threat to email service, causing enormous wasting of human energy, system bandwidth and server storage capacity. Although various anti-spam approaches have been proposed, today spam is still a nightmare for millions of institutes and end-users. The problem is anti-spam filters are shortsighted due to their defensive nature. Consequently, spammers are always one step ahead in the competition. Especially in 2005, spammers introduced a new trick, i.e. image spam, by embedding spam text content into graphical images. The emergence of image spam is a big threat to traditional anti-spam filters which can only handle textual content [20]. Since then, threats continuously exist as spammers keep on appending new tricks to image spam generating processes to defeat anti-spam filtering approaches. For example, randomization techniques [21] are used to evade the signature-based approaches [1, 18]; obscuring techniques [3, 21] are utilized to defeat Optical Character Recognition (OCR) based approaches [4, 9, 18] and low-level image feature based approaches [2, 4, 13, 15]. As a result, single anti-spam approach is no longer effective at detecting diverse real-world image spam.

In order to solve the problem, anti-spam frameworks e.g. [5, 11, 12, 16, 22] are proposed to take advantage of both existing and new anti-spam technologies. With an appropriate combining role, anti-spam frameworks usually achieve higher filtering capacity. However, other factors should also be taken into consideration. Generally a good framework should meet all the following requirements:

DOI : 10.5121/ijcsea.2014.4103

- **Capacity**: The filtering capacity should be strong enough to detect current spamming tricks.
- **Precision**: The filtering accuracy should be high while the error rate should be low especially in terms of FPR (false positive rate, valuable information may lose if legitimate emails being misclassified as spam).
- **Speed**: The computational complexity should be low.
- **Adaptability**: The framework should have the ability to adjust itself to overcome new spamming tricks.

In this paper, we propose an anti-spam framework based on the above requirements, which can be used in both server-side and client-side. The main contributions are summarized as follows:

- This paper proposes a novel clustering filter. By using density-based clustering algorithm, the proposed filter aims to detect spam generated from the same spam template and is robust to image spamming tricks.
- By integrating the clustering filter with a machine learning based filter which contains multiple machine learning classifiers, a combinational framework is proposed to further improve the filtering capacity and precision.
- In order to ensure the long-term efficiency of the proposed framework, incremental learning mechanism is utilized to ensure the framework be able to adjust itself to accommodate future image spam.
- Experiments are carried out to evaluate the effectiveness of the framework and the incremental learning mechanism. The experiment results of two public spam corpora demonstrate the proposed approach achieves high filtering precision.

The rest of the paper is organized as follows. In Section 2, we take a general look at the spamming techniques as well as the current anti-spam filtering approaches. Section 3 introduces the clustering filter. In Section 4, we present the integrated filter. The architecture of the proposed framework and the incremental learning mechanism are discussed in Section 5. The experiment results and evaluation are given in Section 6. Finally, Section 7 concludes the paper.

2. SPAMMING AND FILTERING TECHNIQUES

This section provides a general description of the phenomena of image spam as well as the respective advantages and drawbacks of current anti-spam filtering approaches.

2.1. Image Spam Specification

Spammers take advantage of various spamming tricks to make their image spam more and more difficult to be detected. In order to defeat image spam, recent researches have analyzed the image spamming techniques. In [21], the authors divide the image spam generating process into two steps. The first step is template construction, where spam templates are constructed with the intended text content and illustrations (usually are advertising materials). The second step is randomization, where a large number of spam images can be generated from templates through various randomization techniques, so as to defeat signature-based anti-spam techniques. Besides, a series of image obscuring methods are usually applied during the spam generating processes e.g. text obfuscation, font alteration, rotation, noising, and recoloring [1, 3], which can easily defeat the OCR-based filtering approaches meanwhile causing troubles to the image feature based filtering approaches.

2.2. Anti-Spam Filtering Approaches

Current anti-spam filtering approaches can be divided into three categories i.e. OCR-based approaches, image feature based approaches and combination approaches. Each category is briefly described as follows.

2.2.1. OCR-Based Approaches

OCR is a technique to convert optical characters in images or scanned documents into machine-encoded characters [10]. Since spammers usually embed junk text information into image spam, the OCR technique is utilized to extract text content from images. After that, the extracted text can be checked by traditional text-based filters. The well-known commercial anti-spam tools such as SpamAssassin and Mailcleaner all detect image spam based on OCR technique. However, there are two issues coming along with OCR. First, it is vulnerable to obscuring tricks. Second, it imposes high computational cost. Because of the two issues, anti-spam approaches that rely solely on OCR are no longer effective to detect current image spam.

2.2.2. Image Feature Based Approaches

Since low-level image features are more robust than OCR in image spam detection, diverse image features have been used in image spam filtering approaches, among which the most commonly used features are color, edge, texture and so on [1]. Besides, the image feature based approaches always utilize machine learning techniques [4]. However, one of the issues encountered by image feature based filtering approaches is feature selection. In order to gain high filtering accuracy, it is necessary to appropriately select a representative feature group which should not be too big or too small. The other issue is due to the semantic gap between low-level image features and the high-level concepts implied by the image.

2.2.3. Combination Approaches

Combination approaches are proposed by combining multiple filtering techniques together, so as to reach an overall filtering capacity better than those of any single filtering technique. In [11, 16, 22], combination framework are proposed by integrating OCR-based and image feature based classifiers. In [5, 12, 21], multiple image feature based classifiers are combined together. Even though all of these approaches have achieved better filtering capacity than single technique, none of them take all the aforementioned four requirements into consideration.

3. CLUSTERING FILTER

In this section, we introduce a novel incremental clustering filter, which is a sub-filter of the framework. Generally, Clustering-based filters are robust to obscuring and randomization tricks. Because, spammers always generate a large number of image spam from one template during spam generating process. Since these images are very similar to each other, they can be grouped together by clustering methods, thus being separated from other images. Consequently, the filtering results of clustering-based filters are highly credible, especially for detecting image spam derived from known spam clusters. However, there are two issues with current clustering-based filters [13, 15, 22]. First, these filters can only discover clusters with convex shape. But in large spatial feature space the spam clusters may exist in non-convex shape. Second, most of these filters do not support incremental clustering which is crucial to image spam filtering, because image spam are evolving over time.

3.1. Density-Based Clustering Algorithm

To overcome the drawback of current clustering-based filters, we select density-based clustering algorithm to construct the proposed clustering filter. Density-based clustering algorithm has three advances:

- It can discover clusters with arbitrary shape
- It can discover arbitrary number of cluster with minimal number of input parameters.
- It has good efficiency on large spatial databases with noise (spam feature space usually contains isolated noise points).

The principle of density-based clustering algorithm is to search regions of high density in a feature space. Figure 1 shows an example of density-based cluster in the two-dimensional space. In particular, the points of a cluster are divided into two kinds. (a) Core points are points inside of the cluster (c1 in Figure 1). For each core point of a cluster the neighborhood of a given radius has to contain at least a minimum number of points (*Minpts*), where radius and *Minpts* are two given parameters. (b) Border points are points on the border of the cluster (b1 and b2 in Figure 1). For such points the neighborhood of a given radius has fewer points than number *Minpts*, but must contain at least one core point. The other points outside the cluster are considered as noise (n1 and n2 in Figure 1).

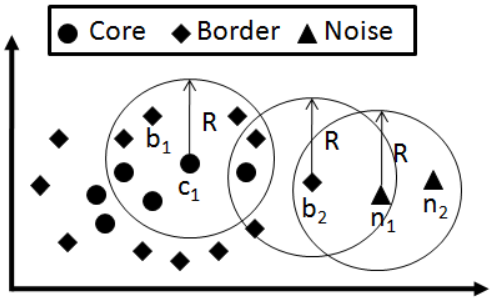


Figure 1. Relation of core, border and noise points in two dimensional space

A plenty of density-based clustering algorithms have been proposed with respective advantages and drawbacks, such as DBSCAN, VDBSCAN, DVBSCAN and so on [17]. DBSCAN can discover clusters with arbitrary shape. VDBSCAN is capable of finding out clusters with varied densities. DVBSCAN is able to handle the density variation within a cluster. Clusters detected by DVBSCAN are separated by sparse region as well as the regions having the density variation. Finally, we choose DBSCAN to build the clustering filter ,because DBSCAN is one of the simplest and fastest algorithms which holds good for large spatial databases. It requires two input parameters, (i) the radius of the cluster (ii) the minimal points required inside the cluster (*Minpts*). Although DBSCAN cannot detect clusters with varied density, it has advantages in time and space complexity, which is a more crucial factor for anti-spam filters (especially for server-side approaches).

3.2. Incremental Clustering

Incremental clustering ensures the clustering filter be able to update itself to new reported image spam. For density-based clusters, adding a new point can result in four cases shown in Figure 2. In the first two cases, the new coming points belong to the existing cluster. The difference is in Figure 2(a) the new point is adjacent to the core points of an existing cluster (namely inclusion),

while in Figure 2(b), the new point is far from the core points but its appearance extends the existing cluster by converting a border point into core (namely extension). Besides, the appearance of new point may also generate a new cluster by converting noise points or itself into core points, as is shown in Figure 2(c). The last case is depicted in Figure 2(d) where the new point becomes a noise.

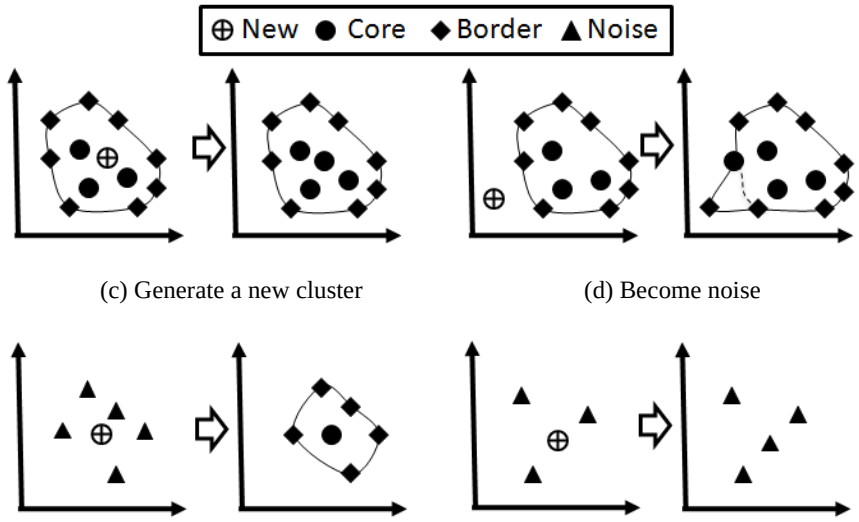


Figure 2. Four cases of incremental clustering

In [7] the authors propose an DBSCAN based incremental clustering algorithm that works in two steps:

Step 1. Compute the means between every core points of clusters and the new coming data. Insert the new data into a particular cluster based on the minimum mean distance. Mark the data as noise or border if it cannot be inserted into any clusters.
Step 2. Form new core points or clusters when noise points or border points fulfill the *Minpts* and *radius* criteria.

We adjust the incremental clustering algorithm to optimize the performance. In particular, a singly-linked list, namely noise list, is maintained for each border and noise point. The noise list records all the neighboring noise points of a particular point. Thus, during incremental clustering process, when non-core points become new core points (depicted in Figure 2(b) and Figure 2(c)), it is no need to recalculate their neighboring noise points (neighboring noise points of a new core point should be marked as border points). Above all, the maintaining of noise list only occupies little memory and does not infect the computational complexity of the original algorithm.

3.3. Filtering Policy

In this work, we propose a filtering mechanism that contains two filtering policies i.e. a fast filtering (FF) policy and a strict filtering (SF) policy. As the name implies, the FF policy emphasizes the filtering speed, while the SF policy concerns more about the filtering capacity. Table 1 lists the differences between the two policies.

Table 1. The two filtering policies

Case / Policy	Inclusion	Extension	New Cluster	Noise
FF	Spam	Ham	Ham	Ham
SF	Spam	Spam	Spam	Ham

In case of FF policy, the filter measures the distance between the incoming image and every core points in the spam feature space. If the distance is within the threshold *radius* (an inclusion case happens, shown in Figure 2(a)), the incoming image will be classified as spam. Otherwise, it will be classified as ham. Since FF policy only takes core points into consideration, the computational complexity is O(m) with the average run time complexity of O(log m), where m is the number of core points. However, the gaining in speed may cause certain extent of reduction in filtering capacity, because the FF policy neglects the information carried by border and noise points.

The SF policy extends the FF policy by further measuring the distance between the incoming image and other non-core points. If the incoming image causes any of the non-core points or itself becoming core (either extension or new cluster is occurs, shown in Figure 2(b), Figure 2(c)), the filter will classify the incoming image as spam. Otherwise, the incoming image will be classified as ham. In the worst case, the SF policy needs to measure the distance between the incoming image and all the points. So the computational complexity degrades into O(n) with the average run time complexity of O(log n), where n is the total number of points in the spam feature space. We evaluate the effectiveness of the two filtering policies in Section 6.

3.4. Cluster Maintenance

As more and more spam are reported by end users, traffic monitors or any other spam detection methods, the spam feature space of the clustering filter will become crowded, which will slow down the filtering speed. In order to guarantee the long-term performance, we add an additional date field namely *latest_date* to denote the activity of each cluster. *latest_date* is initialized to the time when the cluster is created. Each time when a new spam is added into the cluster or a spam is detected by the cluster, *latest_date* will be updated to the current time. Thus, the newer the *latest_date*, the more active the cluster is. Since new coming spam are more likely to be matched by active clusters, the filtering speed can be optimized by regularly ranking the existing clusters in order of *latest_date* to make active clusters taking effect in filtering process before inactive clusters. Besides, another way of optimizing is to remove out-of-date clusters from the feature space. However, this operation should be reversible. Once out-of-date clusters match misclassified spam, they should be added into the feature space once again.

4. INTEGRATED FILTER

Every single anti-spam approach has defects. Considering the clustering filter, it is inefficient to detect spam from unknown spam sources. In order to make up the defect, we propose an integrated filter, which integrates multiple machine learning methods to enhance the filtering capacity. However, how to aggregate the results from multiple classifiers is still an unsolved issue.

We propose a simple but effective voting mechanism. Suppose the integrated filter contains n binary sub-filters. Then, each sub-filter has an output O_i as well as a weight W_i which will be further discussed in Section 5. Thus, the voting result can be summarized as follows:

$$result = \begin{cases} spam, & \sum_{i=1}^{n} O_i W_i \geq \alpha \sum_{i=1}^{n} W_i \\ ham, & \sum_{i=1}^{n} O_i W_i < \alpha \sum_{i=1}^{n} W_i \end{cases}$$

Since the value of O_i is either 1 (spam) or 0 (ham), the summation of weighted outputs $\sum_{i=1}^{n} O_i W_i$ is within the range of [0, $\sum_{i=1}^{n} W_i$]. A arger value means an image is more likely to be spam. α is the filtering threshold used to balance the FPR and the filtering capacity, whose value is within [0, 1]. Set α to a large value may decrease the FPR. For instance, if the value of α is set to be 1, an input image will be classified as spam only if all sub-filters consider it as spam. On the other hand, reducing the value of α can improve the filtering capacity. Actually, the value of α depends on diverse factors, such as the total number of sub-filters, the respective accuracy of sub-filters and even the user preferences. Hence, instead of setting a fixed value of α, we recommend a range [1/2, $\overline{accuracy}$] for α. Here, $\overline{accuracy}$ is the mean value of sub-filters' accuracy which should be higher than 1/2. The accuracy of anti-spam filters is defined as follows:

$$accuracy = \frac{n_{ham \to ham} + n_{spam \to spam}}{N_{ham} + N_{spam}}$$

Here, N_{ham} and N_{spam} denote the total number of ham and spam images respectively. $n_{ham \to ham}$ and $n_{spam \to spam}$ each denotes the number of ham or spam images being classified accurately.

5. PROPOSED FRAMEWORK

In the former sections, we have introduced two filters with respective advantages and drawbacks. In this section, we propose a combination framework which can not only take advantage of both filters' strength but also dampen the drawback of each filter. In addition, incremental learning mechanism is utilized to ensure the long-term filtering efficiency.

5.1. Framework Architecture

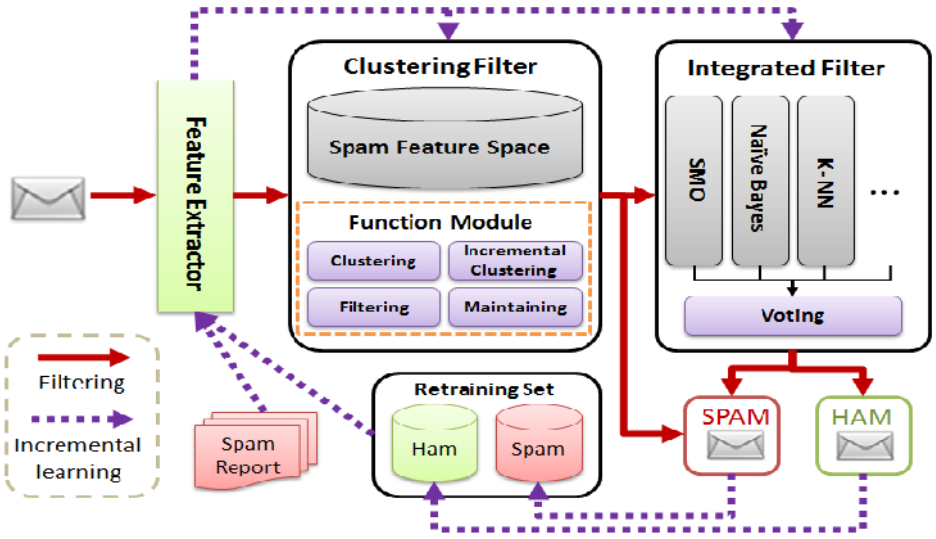


Figure 3. Anti-Spam framework architecture

The proposed framework is depicted in Figure 3. The feature extractor is used to extract low-level image features. During the filtering process, the clustering filter makes the filtering decisions before the integrated filter. All images that are classified as spam will be put into the spam dataset directly. Since the clustering filter is more resistant to obscuring tricks, let it make the filtering decision in advance can improve the filtering precision. The images that have passed through the clustering filter will be further inspected by the integrated filter. Thus, spam from unknown spam sources can also be detected. The combination of the two filters will improve the filtering capacity. Furthermore, this hierarchical framework can outperform the concurrent framework in terms of filtering speed. Finally, emails will be classified as spam once the proportion of contained spam images is above certain ratio (e.g. 20%, ham images are usually mixed up with spam images to confuse the filter).

5.2. Incremental Learning Mechanism

It is worth noting that, image spam are not static but evolving over times. To ensure long-term efficiency, anti-spam filters should have the ability to incrementally learn the characteristics of new image spam.

The clustering filter is capable of constantly updating itself to new image spam. Generally, new image spam come from two resources. One is the spam detected by the integrated filter. The other is the spam detected by other methods such as traffic analysis, sender analysis and end user reports.

As for the integrated filter, to ensure long-term efficiency, sub-filters should be retrained by both old and new image spam. In the proposed framework, all processed images will be stored in the retraining sets. Duplicates should be removed and clustering analysis methods can be utilized to pick up representative images so as to further refine the retraining set. So far, there is still an unsolved issue. That is when to retrain the sub-filters. Generally, a sub-filter requires retraining when its precision drops below a predefined threshold. Each filter's precision is regulated by user feedback. Specifically, when a misclassification has been reported by users, the integrated filter will recalculate the accuracy and FPR for each sub-filter. The integrated filter maintains a weight W_i for each sub-filter, shown as follows:

$$W_i = accuracy_i (1 - FPR_i)^{\beta}$$

here, $accuracy_i$ and FPR_i are respectively the accuracy and FPR of the i^{th} sub-filter. β is a positive integer. Its value should be greater than 1. Since high FPR is the main problem for anti-spam filters, a high value of β may suppress the sub-filters with high FPR. In the beginning, all the weight values are initialized to 1. But over time, weight values will drop gradually. Once the weight value is lower than a predefined threshold, the corresponding sub-filter will be retrained, after that the weight will be reset to 1.

6. EXPERIMENTS AND RESULTS

6.1. Feature Extraction

In our experiment, we extract three kinds of image features based on MPEG-7 standard [6]. The average computing time for feature extraction was 186ms per image on a Pentium Dual 2.0GHz machine. The reasons for selecting the features as well as the detail of each feature are described below.

Color Structure: Color features are usually robust to obscuring tricks such as random noises, rotation, resizing and resolution. Color structure represents an image by both the color distribution and the local spatial structure of the color. It makes up the drawback of traditional color features which cannot represent spatial structure of the color. More importantly, color structure can be calculated very efficiently. It calculates color information in a structuring element of 8×8 pixels that slides over the image. The color values are represented in HMMD color space. The HMMD space used by the Color Structure Descriptor is defined by three components, Hue (the angle specifies one color family from another), Diff (the tone) and Sum (the brightness). A non-uniform quantization is performed on the three components [23]. Finally, each color point is specified by 128 bins. Each bin is quantified to 8 bits.

Edge Histogram: Edge features are among the most efficient image features in image spam detection because they can reveal the characteristics of the boundary between foreground (illustration and text) and background of spam images. Edge histogram is an efficient edge feature which represents the spatial distribution of five kinds of edges, i.e. horizontal, vertical, 45ºdiagonal, 135ºdiagonal and non-directional edges. An image is divided into 4×4 sub-images without overlapping. Each sub-image is assigned five bins which represent the relative frequency of occurrence of the five kinds of edges in the corresponding sub-image. As a result, edge histogram contains 80 bins.

Homogenous Texture: Texture features are usually useful to distinguish computer-generated image e.g. spam, from natural images [19]. Homogenous texture is an effective texture feature. It provides a quantitative representation of the image texture based on Gabor filters and is also robust to many obscuring tricks such as random line, frame, different font type and so on. In particular, an image is filtered with a bank of Gabor filters having 6 orientations and 5 scales. The first and the second moments of the energy in the frequency domain in the corresponding sub-bands are calculated. Together with the mean brightness and standard deviation of the image pixels, homogenous texture contains 62 bins.

However, the values of the three features do not have unified range. To prevent features with large range from outweighing features with smaller range in distance measurement, we normalized the extracted features by scaling all their values in the range of [0, 1]. Finally, we used L2 distance to measure the distance between two feature vectors.

6.2. Thresholds Determination

The clustering algorithm requires two predefined thresholds: *Minpts* and *radius*. We present a new method to determine the thresholds based on the following observation. Given a training set, it is easy to section out all the isolated images that are dissimilar with all the other images artificially. The isolated images can be marked as noise in advance and then be used to evaluate the clustering precision. The precision is defined as:

$$precision = \frac{n_{correct}}{N + n_{detect} - n_{correct}}$$

N is the real number of noise points, which is a const. n_{detect} denotes the detected number of noise points. $n_{correct}$ represents the number of real noise points that have been correctly detected. The precision will increase as the value of n_{detect} and $n_{correct}$ become closer to N. Therefore, high precision indicates high clustering accuracy of noise points, which in turn reflects high accuracy of none-noise points as well.

The maximum possible range of *radius* and *Minpts* should be provided before invoking the proposed method. In order to select the range, we selected a testing dataset containing 310 typical spam images. In the experiment, the lower limit of *radius* was set 0.9 based on the observation that the minimal distance between two spam generated from the same template (which should be in the same cluster) is approximately equal to 0.9. On the other hand, the upper limit was set 3.1 as the minimal distance between two spam generated from different templates (which should be in different clusters) is approximately equal to 3.1. The range of *Minpts* was set from 2 to 8 based on empirical estimate to ensure the range is big enough.

The method runs in the following steps to search the optimal thresholds:

- **Step 1:** Divides *radius* into values of certain interval (with initial value of 0.2) within its range. Generates all the possible combinations of the two thresholds;
- **Step 2:** Clusters the training set with each pair of thresholds and selects the top-5 pairs having the highest precision;
- **Step 3:** Reset the range of *radius* by linking the values of *radius* from the top-5 pairs with their adjacent ranges of length interval. Multiply the interval by 0.25;
- **Step 4:** Repeat the former steps until the interval is less than 0.01;

Finally, the pair of thresholds achieving the highest precision will be selected. In our experiment, the highest precision was 99.4% with *Minpts* being 4 and radius being 2.2375. In addition, we constructed the integrated filter with three machine learning classifiers implemented in WEKA [14], i.e. Naïve Bayes, K-NN and SMO (using RBF kernel). We choose these classifiers because they are robust, suitable for high dimensional features and simple to use. Parameters of the classifiers are optimized with the help of WEKA meta-classifier CVParameterSelection. On the other hand, the incremental learning threshold α was set as 0.7 and the parameter β was set as 2.

6.3. Dataset and Experimental Process

The dataset used in the experiment came from two public spam corpora[1]. In order to ensure credibility, we preprocessed the corpora by removing duplications, invalid images and images smaller than 128×128. Finally, we got 12800 images including 6000 ham and 6800 spam. To measure the size of the experimental images, we divided the experimental images into six groups according to the size of the image measured by megapixel. Figure 4 shows the ratio of each group of experimental images. The 12800 images are randomly divided into eight equal size subsets.

[1] *http://www.cs.northwestern.edu/~yga751/ML/ISH.htm*
http://www.cs.jhu.edu/~mdredze/datasets/image_spam/

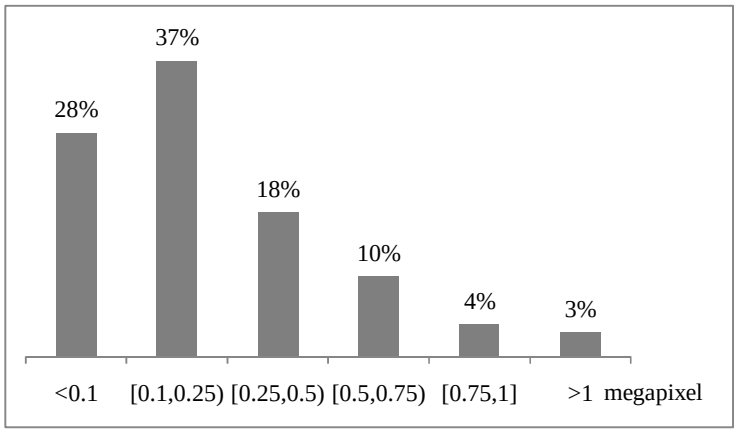


Figure 4. The ratios of experimental images in different size groups

The experiment contains three steps:

- The first step is to evaluate the effectiveness of the incremental learning mechanism. We randomly selected one subset from eight subsets as testing set. The remaining seven subsets were used as incremental training sets. We incrementally trained the framework, each time with a new training set until all seven subsets had been used. After each time's training, testing set was utilized to evaluate the performance.

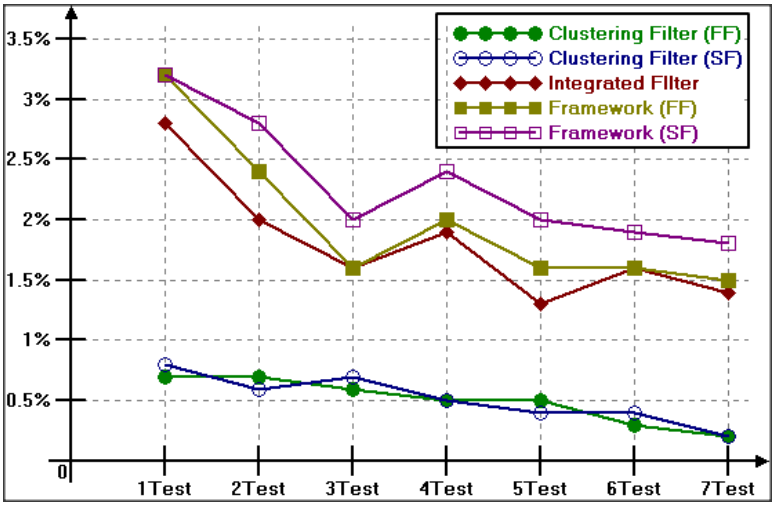
- The second step is to evaluate the effectiveness of the framework as well as each sub-filter. In the second step, eight-fold cross-validation was used to evaluate the proposed approach. Since the clustering filter contains two filtering policies, we evaluated the clustering filter and the framework under two policies respectively in terms of accuracy, FPR and speed.
- In the third step, we compared the performance of the proposed approach with another state-off-the-art anti-spam approach namely HAF (hierarchical anti-spam framework), proposed in [16]. In particular, HAF is also a typical combinational anti-spam framework. However, the filtering principles of HAF is based on obscuring detection and OCR techniques while the approach in this paper is based on clustering and machine learning techniques. In order to compare the performance between the proposed approach and HAF, we evaluate HAF with the same dataset mentioned before. Finally we compared the false negative rate, false positive rate, accuracy, and filtering speed of the two approaches.

6.4. Evaluation

The experiment results of the first step are shown in Figure 5. Figure 5 (a) depicts the FPR of each filter and Figure 5 (b) depicts the filtering accuracy. FPR reflects the error rate of ham being misclassified as spam and accuracy denotes the overall filtering efficiency. It can be seen that the effect of incremental learning is obvious. FPRs of the five tested approaches decline gradually. For instance, the FPRs of framework under FF and SF policies drop 1.7% and 1.4% respectively. On the other hand, as a result of the incremental learning mechanism, the accuracies increase significantly. For example, the accuracies of framework under FF and SF policies increase 3.0% and 2.8% respectively.

The average performances of eight-fold cross-validation are shown in Table 2. The clustering filter has relatively low FPR with all the values below 1%. Obviously, the framework either using FF policy or SF policy outperforms single filters in terms of filtering accuracy. Comparing the performances of the framework under two policies, the FF policy has advantages over the SF policy in terms of FPR and filtering speed while the SF policy achieves higher accuracy. In particular, the FPR of the FF policy is 0.2% lower than the SF policy and the average filtering speed per image is 13.5ms faster than the SF policy. Nevertheless, the accuracy of SF policy is 0.8% higher than the FF policy. In conclusion, both policies have achieved high filtering performance but with different emphasis.

(a) The FPR of each test



(b) The accuracy of each test

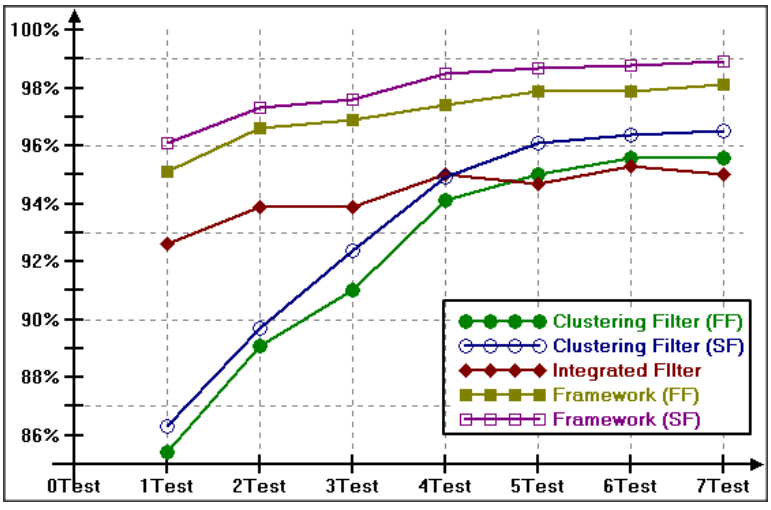


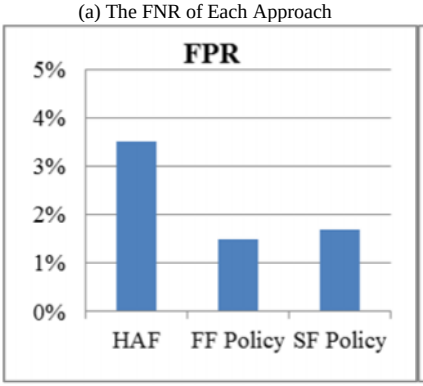
Figure 5. Experimental results of incremental learning

Table 2. The average performance of the proposed framework

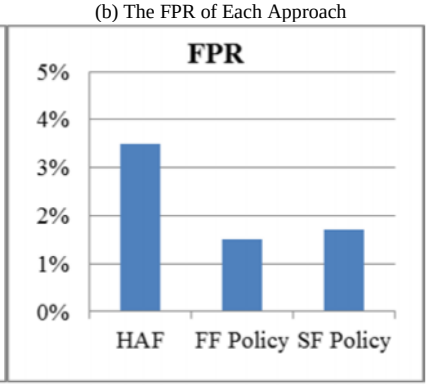
	FPR	Accuracy	Speed (per image)
Clusterting Filter (FF)	0.2%	95.6%	55.2 ms
Clustering Filter (SF)	0.3%	96.7%	68.7 ms
Integrated Filter	1.4%	95.0%	29.3 ms
Framework (FF)	1.5%	98.2%	84.5 ms
Framework (SF)	1.7%	99.0%	98.0 ms

The experiment results of the third step are shown in Figure 6. Figure 6 (a) compares the FNR (false negative rate) of the proposed approach with HAF. FNR reflects the error rate of spam being misclassified as ham. The proposed approach achieves lower FNR than HAF especially by using FF policy. It means the proposed approach has advantages than HAF in spam image detection. Figure 6(b) shows the FPR of the proposed approach and HAF. FPR reflects the error rate of ham being misclassified as spam, which may cause the loss of valuable information. Denoted by the figure, the proposed approach achieves lower FPR than HAF, which means the risk to use the proposed approach is lower than HAF. Figure 6 (c) shows the filtering accuracy of the tested approaches, from which we can see, the proposed approach achieves higher accuracy than HAF, especially by using SF policy. Figure 6 (d) denotes the filtering speed (per image). As we can see, the filtering speed of the proposed approach either FF policy or SF policy is faster than HAF.

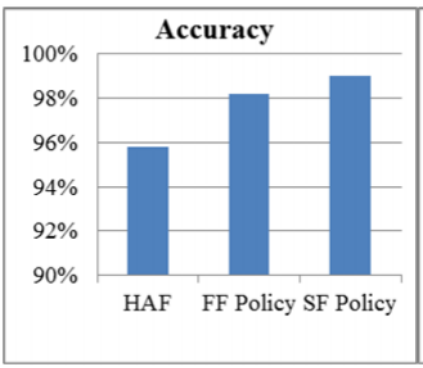
(a) The FNR of Each Approach



(b) The FPR of Each Approach



(c) The Accuracy of Each Approach



(d) The Filtering Speed Per Image of each

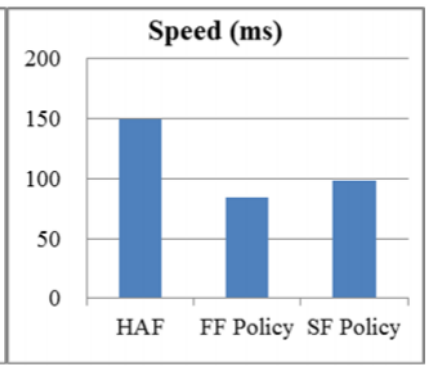


Figure 6. Performance Comparison

In order to evaluate the proposed approach with HAF based on more various testing criteria, we divided the 6800 testing spam images into five categories, i.e. C1~ C5. Table 3 represents the description of the five categories as well as the proportion of each category.

We compared the spam detection rate of the tested approaches on the five spam categories. Spam detection rate was obtained by letting the number of correctly detected spam divide by the total number of spam in the corresponding category. The result is shown in Table 4. Obviously, the SF policy has advantages than HAF in respect of spam detection rate on all five kinds of image spam, especially for detecting spam that contain obscured text contents with complex background (C4). This is because the effect of the OCR model in HAF is affected by highly obscured images. Considering the FF policy, even though its spam detection rates are not as high as the SF policy, it has advantages than HAF in detecting spam of C2, C3, C4 categories. It is worth noting that, these three kinds of spam occupy 88% of all image spam. The only category, that the proposed framework using FF policy showing lower spam detection rate than HAF, is C1 (spam contain relatively clear text contents) which only occupies 8% of all image spam.

Table 3. The Five Categories of Spam Images

Category	Description	proportion
C1	Images that contain relatively clear text contents.	8%
C2	Images that contain text contents with complex background.	10%
C3	Images that contain obscured text contents.	74%
C4	Images that contain obscured text contents with complex background.	4%
C5	Images that contain complex background or illustrations but without text content.	4%

Table 4. The Spam Detection Rate of the Tested Approaches

	C1	C2	C3	C4	C5
HAF	99.1%	98.0%	96.2%	66.7%	93.3%
FF	96.3%	99.2%	98.4%	94.0%	93.3%
SF	99.6%	99.4%	99.7%	98.7%	98.3%

7. CONCLUSIONS

In this paper, we propose an incremental learning based framework for image spam detection. The proposed framework combines two sub-filters, i.e. a clustering filter and an integrated filter. The clustering filter relies on density based clustering algorithm and is robust to spamming tricks. Besides, two filtering policies are presented with respective emphasis. The Fast Filtering policy has advantage in filtering speed while the Strict Filtering policy has stronger filtering capacity. The integrated filter integrates multiple state-off-the-art machine learning classifiers and is capable of further enhancing the filtering capacity. Finally, incremental learning mechanism is utilized to ensure the proposed framework be capable of incrementally learning feature from new image spam.

The experimental results on two public spam corpora demonstrate the effectiveness of the proposed framework and the incremental learning mechanism. Consequently, the proposed framework achieves high filtering precision and filtering speed. For Fast Filtering policy, the filtering accuracy is 98.2% and the FPR is 1.5%. For Strict Filtering policy, the filtering accuracy reaches 99.0% and the FPR is 1.7%. In addition, the average processing time per image is less than 100ms. Moreover, the comparison of the proposed approach with HAF denotes the proposed approach has advantages than HAF in various testing criteria.

In the future, we plan to take other clustering algorithms into consideration for addressing the problem of image spam. Moreover, we will further optimize the proposed framework and compare its performance with other state-of-the-art proposals.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2008578) && This research was funded by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

REFERENCES

[1] A. Attar, R.M. Rad & R.E. Atani (2011) "A Survey of Image Spamming and Filtering Techniques", *Artificial Intelligence Review*.

[2] Al-Duwairi, I. Khater & O. Al-Jarrah (2011) "Texture Analysis-Based Image Spam Filtering", Proc. *6th Int. Conf. on Internet Technology and Secured Transactions*, Abu Dhabi, UAE, pp.288-293.

[3] B. Biggio, G. Fumera, I. Pillai & F. Roli (2007) "Image Spam Filtering by Content Obscuring Detection", Proc. *4th Int. Conf. on Email and Anti-Spam*, Mountain View, CA.

[4] E. Blanzieri & A. Bryl (2008) "A Survey of Learning-Based Techniques of Email Spam Filtering", *Artificial Intelligence Review*, Bol. 29, No. 1, pp. 63-92.

[5] B. Byun, C.H. Lee, S. Webb, D. Irani & C. Pu (2009) "An Anti-spam Filter Combination Framework for Text-and-Image Emails through Incremental Learning", Proc. *6th Int. Conf. on Email and Anti-Spam*, Mountain View, CA.

[6] M. Bastan, H. Cam, U. Gudukbay & O. Ulusoy (2010) "BilVideo-7: An MPEG-7 Compatible Video Indexing and Retrieval System", IEEE *MultiMedia*, Vol. 17, No. 3, pp. 62-73.

[7] S. Chakraborty & N.K. Nagwani (2011) "Analysis and Study of Incremental DBSCAN Clustering Algorithm", *International Journal of Enterprise Computing and Business Systems*, Vol. 1, No. 2.

[8] M. Ester, H.P. Kriegel, J. Sander & X.W. Xu (1996) "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proc. *2nd Int. Conf. on Knowledge Discovery and Data Mining*, pp.227-231.

[9] G. Fumera, I. Pillai & F. Roli (2006) "Spam Filtering based on the Analysis of Text Information Embedded into Images", *The Journal of Machine Learning Research*, Vol. 7, Dec, pp. 2699-2720.

[10] H. Fujisawa (2008) "Forty years of research in character and document recognition---an industrial perspective", *Pattern Recognition*, Vol. 41, No. 8, pp. 2435-2446.

[11] F. Gargiulo & C. Sansone (2008) "Combining Visual and Textual Features for Filtering Spam Emails", Proc. *19th Int. Conf. on Pattern Recognition*, Tampa, USA, pp.1-4.

[12] Y. Gao, A. Choudhary & G. Hua (2010) "A Comprehensive Approach to Image Spam Detection: from Server to Client Solution", IEEE *Trans. Information Forensics and Security*, Vol. 5, No. 4, pp. 826-836.

[13] S. Gao, C.C. Zhang & W.B. Chen (2011) "Identifying Image Spam Authorship with Variable Bin-width Histogram-based Projective Clustering", Proc. IEEE *Int. Conf. on Multimedia and Expo*, Barcelona, Spain, pp. 1-6.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann & L.H. Witten (2009) "The WEKA Data Mining Software: An Update", ACM *SIGKDD Explorations Newsletter*, Vol. 11, No. 1, pp. 10-18.

[15] Y. He, W.G. Man & H.B. He (2011) "Incremental Clustering-based Spam Image Filtering using Representative Images", Proc. *2011 Int. Conf. on System Science, Engineering Design and Manufacturing Informatization*, GuiYang, China, pp. 323-327.
[16] X.M. Li & U.M. Kim (2012) "A Hierarchical Framework for Content-based Image Spam Filtering", Proc. *8th Int. Conf. on Information Science and Digital Content Technology*, Jeju, Korea, pp. 149-155.
[17] M. Parimala, Lopez Daphne & N.C. Senthilkumar (2011) "A Survey on Density Based Clustering Algorithms for Mining Large Spatial Databases", *International Journal of Advanced Science and Technology*, Vol. 31, No. 5, pp. 59-66.
[18] H. Stern (2008) "A Survey of Modern Spam Tools", Proc. *5th Int. Conf. on Email and Anti-Spam*, Mountain View, CA.
[19] C.T. Wu, K.T. Cheng, Q. Zhu & Y.L. Wu (2005) "Using Visual Features for Anti-Spam Filtering," Proc. *IEEE Int. Conf. on Image Processing III*, Genoa, Italy, pp. 501-504.
[20] L. Zhang, J.B. Zhu & T.S. Yao (2004) "An evaluation of statistical spam filtering techniques", ACM *Trans. Asian Language Information Processing*, Vol. 3, No. 4, pp. 243-269.
[21] Z. Zhang, W. Josephson, Q. Lv, M. Charikar & K. Li (2007) "Filtering Image Spam with Near-Duplicate Detection", Proc. *4th Int. Conf. on Email and Anti-Spam*, Mountain View, CA.
[22] C.C. Zhang, W.B. Chen, X. Chen, R. Tiwari, L. Yang & G. Warner (2009) "A Multimodal Data Mining Framework for Revealing Common Sources of Spam Images", *Journal of Multimedia*, Vol. 4, No. 5, pp. 313-320.
[23] D. Messing, P. van Beek & J.H. Errico (2011) "The MPEG-7 Colour Structure Descriptor: image description using colour and local spatial information", Proc. IEEE *Int. Conf. on Image Processing*, Thessaloniki, Greece, October, pp. 670-673.

Authors

Li Xiao Mang received the BSc degree in information and communication engineering from Sungkyunkwan University. His research interests include data mining.

HaRim Jung received his B.S. degree in Computer Science from Kwangwoon University, Seoul, Korea, in 2004. He received his M.S. and Ph.D. degrees in Computer Science and Engineering from Korea University, Seoul, Korea, in 2007 and 2012, respectively. Currently, he is a research fellow at the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea. His research interests include location-based services, spatial data management in mobile/pervasive environments and spatial big data management.

Hee Yong Youn received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 1977 and 1979, respectively, and the Ph.D. degree in computer engineering from the University of Massachusetts at Amherst, in 1988. Currently he is a Professor of the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea, and the Director of the Ubiquitous Computing Technology Research Institute. His research interests include distributed and ubiquitous computing, system software and middleware, and RFID/USN.

Ung-Mo Kim received the B.E. degree in Mathematics from Sungkyunkwan University, Korea, in 1981 and the M.S. degree in Computer Science form Old Dominion University, U.S.A. in 1986. His Ph.D. degree was received in Computer Science from Northwestern University, U.S.A., in 1990. Currently he is a full professor of School of Information and Communication Engineering, Sungkyunkwan University, Korea. His research interests include data mining, database security, data warehousing, GIS and big data.