Spam Image Clustering for Identifying Common Sources of Unsolicited Emails

Chengcui Zhang*
University of Alabama at Birmingham, USA

Xin Chen
University of Alabama at Birmingham, USA

Wei-Bang Chen
University of Alabama at Birmingham, USA

Lin Yang
University of Alabama at Birmingham, USA

Gary Warner
University of Alabama at Birmingham, USA

ABSTRACT

In this paper, we propose a spam image clustering approach that uses data mining techniques to study the image attachments of spam emails with the goal to help the investigation of spam clusters or phishing groups. Spam images are first modeled based on their visual features. In particular, the foreground text layout, foreground picture illustrations and background textures are analyzed. After the visual features are extracted from spam images, we use an unsupervised clustering algorithm to group visually similar spam images into clusters. The clustering results are evaluated by visual validation since there is no prior knowledge as to the actual sources of spam images. Our initial results show that the proposed approach is effective in identifying the visual similarity between spam images and thus can provide important indications of the common source of spam images.

Keywords: spam image; clustering; computer forensics; botnet; cybercrime; data mining

INTRODUCTION

Spamming is a problem that affects people all over the world. Spam is an unsolicited email which has been sent to many people. There can be legal spam, where the sender gave proper contact information and also has an option to no longer receive the messages. However, in almost all situations, spam is illegal. It is an unsolicited mail that the recipient did not ask to receive and did not give the sender permission to send. Spam falsifies the sender information to prevent anyone from finding out where it has been sent from. Botnets are machines that keep on sending spam.

Today, botnets are the main choice for cyber criminals who seek to conceal their identities by using third-party computers as vehicles for their crimes (www.cnn.com/2007/TECH/11/29/fbi.botnets). The FBI has identified at least 2.5 million unsuspecting computer users who have been victims of botnet activities (www.cnn.com/2007/TECH/11/29/fbi.botnets). Spam sometimes attempts to sell a product, convey some messages, or they might also try to trick the recipient to become infected, or attempt to lure them into visiting a website that can infect them.

Spam can cause a lot of problems to internet users. More than 90% of the emails sent on the internet are spam. Billions of dollars of counterfeit software, electronics, as well as shoes, watches, etc., are being sold because of spam advertisements. In this way, huge financial loss occurs to the companies. Spam emails, claiming to be from banks, might also lure users to give their usernames and passwords. Besides software piracy and viruses, spam is also the primary means of phishing and identity theft. Therefore, spam email analysis is one of the most important topics in cyber security. The most effective way of controlling spam emails at the moment is filtering (Carreras & Mrquez, 2001; Clark, Koprinska, & Poon, 2003; Drucker, Wu, & Vapnik, 1999; Sanpakdee, Walairacht, & Walairacht, 2006). However, filters can only differentiate spam emails from non-spam emails but cannot tell the origins of spam. In order to hide their origins, escape detection and spam filter analysis, and to conceal the fact that there are relatively few organizations creating the vast majority of these unsolicited emails, criminals use a variety of intentional obscuring techniques. For example, one of the techniques is to present text primarily as an image, to avoid traditional computer-based filtering of the text. Spam images are sent for two reasons: 1) for advertisement purposes; 2) to hide the textual contents of an email from spam filters. Having no words in the message will not allow spam filters to understand the nature of the message.

Spam images are harder to detect than text spam. Spam images are created when text is embedded into images and content obscuring technologies are used to defeat spam blocking techniques. Spammers use certain methods to defeat traditional anti-spam technologies such as fingerprinting (e.g., md5 (Rivest, 1992)), OCR, and URL blocklist.

1. A text can be embedded in an image which appears as normal text to the recipient but the spam blocking technologies will never be able to “see” the text as it is actually an image.
2. Spammers vary the space between words and lines and also add random speckles to make messages look different to different recipients, though all of them have the same text. By this way, they evade fingerprinting technology such as md5 (Rivest, 1992) by making the images appear unique to standard spam analysis.
3. Use of different colors and varying font size makes it impossible for OCR techniques to find out spam. Also, splitting up one word into two halves with a gap in between deceives OCR techniques.
4. Botnets are also becoming efficient and they can produce a large number of random images within a short time.

In order to stop unsolicited spam emails, we should trace the origins of spams and bring down the servers as well as those used to send spams. In this process, law enforcement offices shall be actively involved as spam propagating is also a legal issue. The goal of this paper is to facilitate this process in providing scientific proof to the source of spams. We regard these spam images as a valuable clue for identifying the origins of spams. This paper is dedicated to the analysis and clustering of spam images based on their visual characteristics. Through clustering, spam images are grouped together. Each cluster contains spam images whose visual effects resemble each

other in the cluster, indicating common origins/sources of those images, i.e., they are created from the same template hence by the same spammer.

There are relatively few works in spam image identification (Byun, Lee, Webb, & Pu, 2007; Mehta, Nangia, Gupta, & Nejdl, 2008; Wu, Cheng, Zhu, & Wu, 2005). All these works address the image spam filtering problem. For example, Byun et al. (2007) proposed a classification method to model and identify spam images. McAfee, an Internet security vendor, also provides image spam filtering functions in its product. The main purpose of these works is to separate spam images from non-spam images thus to perform filtering functions. Visual features, such as color distribution, color heterogeneity, conspicuousness (some contrast feature), and self similarity (repetitive patterns), are used in training the classifiers/filters. In this study, we go one step further to track the source of the spam distributors based on spam image clustering, i.e., if two spam emails have similar visual content, visual layout, and/or editing styles, then they are likely related. This can be used as a strong evidence base to identify and validate spam clusters or phishing groups for the purposes of cybercrime investigation. For example, an approach (Chun, Sprague, Warner, & Skjellum, 2008) was proposed that used clustering techniques to form relationships between email messages and group them into spam clusters. Clusters were evaluated using a visual inspection method. A routine was developed to fetch and save a graphical image, or thumbnail, of the appearance of each destination website. Where the resultant collection of website images from a single cluster was visually confirmed to be the same by sorting the resultant webpage images, a high confidence was placed upon the integrity of the cluster. Our proposed method can not only automate this visual validation process, but link visual similarity directly to the presence of spam clusters.

The proposed spam image clustering algorithm first extracts visual features from images and then performs the clustering. There are four steps in the feature extraction:

- Foreground Extraction – this step separates foreground image content from the background. The foreground image content can be further classified into two categories: texts and picture illustrations. Since most images we collected are advertisements with text areas in them, we first separate these text areas through Optical Character Recognition (OCR). The rest of the foreground areas are picture illustrations. In the following steps, we extract features from these two types of foreground objects separately.
- Foreground Text Layout Analysis – For efficiency purposes, a spam originator often reuses the same editing template to embed spam texts in the images. Images generated this way usually have similar text layout but different background and/or slightly different spatial placement of text blocks. Thus, the text layout information is an important indication of the editing style of spam originators. In this study, we analyze the text areas in spam images and measure the similarity of text layouts between each pair of images.
- Foreground Picture Illustration Analysis – It often happens that in advertising the same product, a spammer tends to use the same picture illustration. However, unlike texts, it is not very efficient to change the content of image illustrations. Some minor editing on the images such as changing image size is the most commonly attempted by the spammers. Therefore, similar foreground picture illustrations may also indicate that they are from the same spammer or the same phishing group. We therefore perform foreground illustration matching based on the SIFT (Lowe, 2004) method (Scale-Invariant Feature Transform), which is a robust method in matching two distorted yet similar images.
- Background Texture Analysis – When editing spam images, it is probably the easiest to change its background color to make it unique. Even created from the same template, the

background colors (and sometimes even the foreground texts) may be different. Thus, color similarity cannot be treated as an important indication of common templates. Instead, we first convert the image background into grayscale. We further find that, although different in color, the background texture features of images created from the same template tend to have less variation. Therefore, in this paper, we analyzed the "homogeneity" and the "orientation" texture features of image backgrounds and found that with our currently colleted spam images, "orientation" textures can better distinguish among different templates than "homogeneity".

Since we do not have any prior knowledge as to the number of possible spammers or templates hence clusters, in this study we use an agglomerative clustering method to build a hierarchical cluster tree. Links in the tree are evaluated in terms of their consistency, and inconsistent links are cut off from the final clusters.

In the rest of the paper, the foreground extraction method is introduced in Section 2. Sections 3 and 4 analyze foreground texts and picture illustrations. Section 5 analyzes background texture and Section 6 introduces the clustering mechanism. Experimental results are presented in Section 7 and Section 8 concludes the paper.

FOREGROUND EXTRACTION

As mentioned earlier, the visual content of a spam image provides an important clue to identifying spam clusters. Two spam images are said to visually resemble each other if they have similar text layout, and/or similar foreground picture illustration, and/or similar background textures. Hence, there is a need to distinguish foreground objects from the background. To recognize foreground objects in the spam image, we first separate text areas through Optical Character Recognition (OCR). This is achieved by adopting the Microsoft Office Document Imaging (MODI) to identify recognizable texts in the spam images. MODI returns the recognized texts and their bounding rectangles. The coordinates of the bounding rectangles infer the location of each recognized word in the image, and thus, can be used in the subsequent text layout analysis. We exemplify the bounding rectangles of recognized words in the next section.

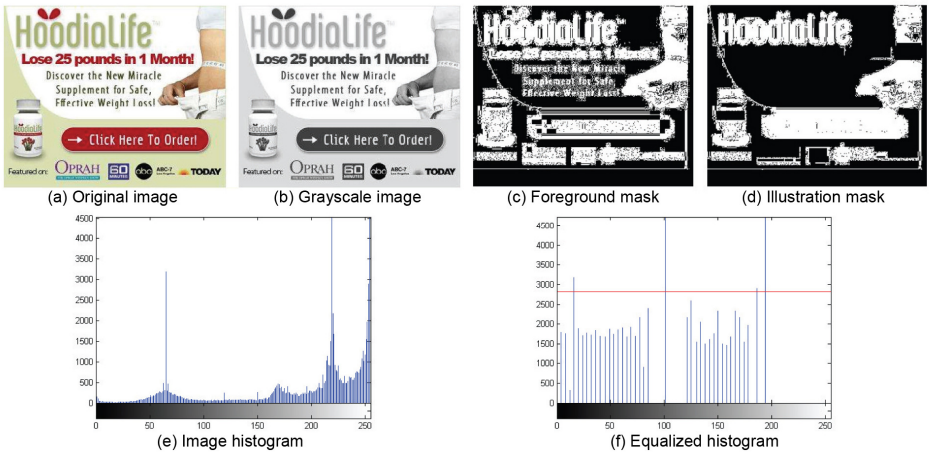
Foreground picture illustrations are then extracted after the text extraction. Picture illustrations can be thought as sub-images in the spam image. To extract picture illustrations from the background, we notice that typically, these sub-images are full of variety in their visual appearance, and thus, difficult to characterize them with any fixed set of visual features. On the contrary, the background is generally composed of a pure color base or computer-generated textures, and has relatively more uniformity than illustrations. Hence, instead of finding illustration areas in a spam image directly, we obtain the illustration areas by removing the background in that image. However, it is not a trivial task since random noise and textures were often added on purpose to increase the background randomness and variations. Hence, we cannot use a single threshold value on visual homogeneity to separate the foreground objects from the background. In this paper, we propose a simple yet effective method to differentiate foreground illustrations from background. The proposed method is based on the following two assumptions. The first assumption is that the spam images must have sufficient foreground/background contrast to ease the reading of their recipients, which is usually the case as indicated by Byun et al. (2007). More specifically, the intensity values of foreground and background must have significant difference. The second assumption is that the background area occupies a significant portion of an image, which is often the largest or at least comparable to foreground illustrations. Also

because background usually demonstrates more uniformity than foreground, background pixels tend to cluster together in the pixel intensity histogram while foreground pixels demonstrate a wide range of intensities.

First, a color image is converted to its corresponding grayscale image as shown in Figure 1(a) and (b). According to the first assumption, the foreground/background contrast can be preserved even after converting it to an intensity image. Histogram equalization is then applied to the image to enhance the contrast. The equalized histogram may have empty bins around peaks since histogram equalization conceptually spreads out the most frequent intensity values into adjacent empty bins, making the histogram a uniform distribution (Burger & Burge, 2007). We demonstrate the histogram equalization in Figure 1(e) and (f).

Based on our second assumption, background pixel intensities usually have a relatively smaller range than that of the foreground and thus correspond to high frequency bin(s) in the equalized histogram. Hence, we first calculate the average frequency of non-empty bins. The empty bins are ignored since they are virtually filled with high frequency values. For all bins with frequency higher than the average, we considered them as corresponding to the intensity values of the background. The red line in Figure 1(f) represents the average frequency, and the black pixels in Figure 1(c) show the identified background pixels. The white areas in Figure 1(d) correspond to the picture illustrations after the removal of the text areas identified by OCR from Figure 1 (c).

Figure 1. (a) the original image, (b) the converted grayscale image, (c) the foreground/background mask, (d) the illustration mask, (e) the original image histogram, and (f) the equalized histogram



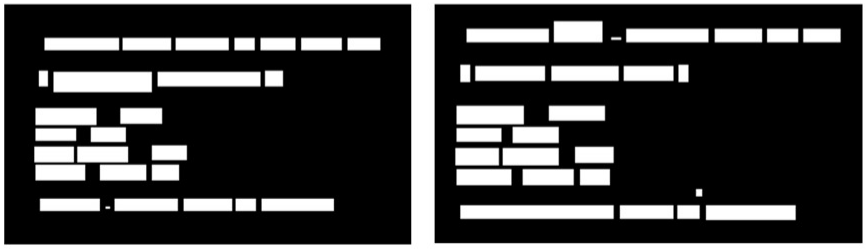
FOREGROUND TEXT LAYOUT ANALYSIS

After the foreground extraction, text blocks contained in original spam images are singled out. The words in two advertising spam images are not necessarily the same when the two spam images are trying to sell different things. However, it is highly possible that a spammer uses the same text layout template in generating different advertisements by only changing the wordings for different products. For example, in Figure 2(a), two spam images have different text contents. However, their corresponding text layouts in Figure 2(b) are very similar. Similar text layouts

may indicate spam images from common origins. Therefore, instead of the exact wording in the texts, we emphasize on the analysis of the text block layouts.

Figure 2. (a) Sample spam images with different text content yet similar text layout. (b) The text block masks of the images in Figure 2(a)

(a)

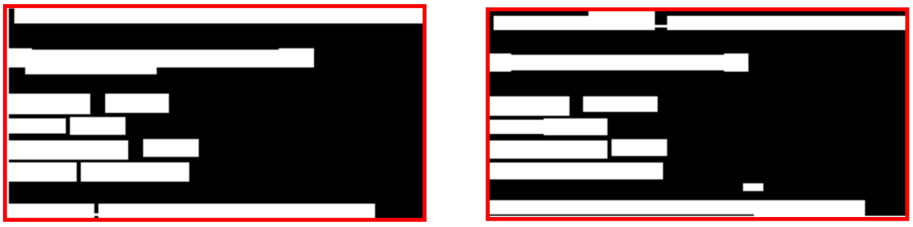


(b)

In this section, we will use the sample images in Figure 2 to illustrate our text layout analysis method.

1. Bounding Box Extraction – The first step in text layout analysis is to extract the minimum bounding box of the whole text area in each image.

Figure 3. Dilation



2. Dilation – We notice that two text layouts may look similar in their general layout yet their word and space distributions are very likely to be dissimilar. This is especially true when different wordings are used in the text as shown in Figure 2(b). If we directly compare the text layout masks, noises will be introduced by different word length, line spacing and word positions in each text line. We alleviate this problem by coarsening the text area. In doing so, we try to connect words in one line if they are only separated by a small space. The method we used to coarsen the text blocks is called dilation. For each pixel in the bounding box, if it is "1", the m pixels on its right and m pixels on its left are also set to 1. In this way, small spaces are "closed" and therefore ignored. Only the general layout of the whole text area will be considered in the analysis. The resultant text bounding boxes and the dilated text areas for Figure 2(b) are shown in Figure 3.

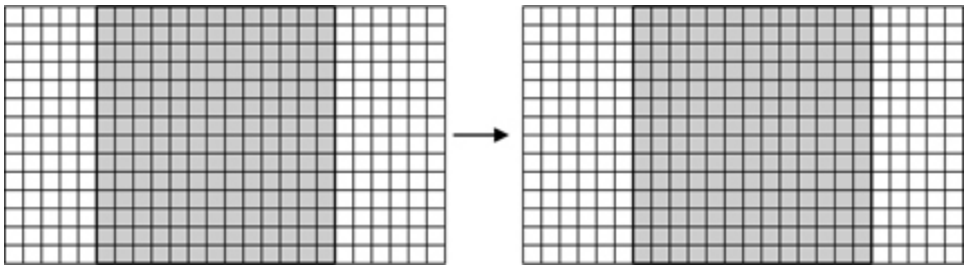
3. Scaling – Text areas from different spam images are usually not of equal size so that we cannot directly compare them. To compare two text layouts, we first need to normalize them. A common way is to down-sample the larger text area, bounded by its minimal bounding box, to the same size of the smaller text area. However, this method may cause the larger text area to be skewed since the aspect ratio of the two images may not be the same. Therefore we only resize the larger text area so that its length is the same as that of the smaller text area. However, the original aspect ratio of its length and width is preserved. Therefore, the two text areas in comparison can have different widths.

4. Similarity Calculation – After resizing, we superimpose the text area with the shorter width on the one with the longer width and conduct the pixel-wise comparison. Then we slide the smaller text area one step at a time and repeat the comparison. This process is illustrated in Figure 4. The grids in Figure 4 represent pictures with their pixels. The smaller text area is represented by dark gray grids. Each time we compare two text areas, we calculate their distances by the following formula:

$$layout(I_1, I_2) = \frac{\sum_{i,j} (I_1(i, j) - I_2(i, j))}{l_{small} \times w_{small}} \quad (1)$$

where $I_1(i, j)$ and $I_2(i, j)$ are the corresponding pixel values at the corresponding position (i, j) of the two text areas. Here the value of a pixel is either 1 (white: text pixel) or 0 (black: non-text pixel). l_{small} and w_{small} are the length and width of the smaller text area. A series of distances are thus calculated by sliding the smaller text area over the larger one. The minimum value of the distances is used to represent the distance between the two text areas i.e., the distance of the two text layouts.

Figure 4. Superimpose the smaller text area on the larger text area and slide it over the larger one to find the best match



FOREGROUND PICTURE ILLUSTRATION ANALYSIS

Almost identical illustrations contained in spam images are strong indications that they originate from the same source. However, now that we are measuring sub-regions of the spam images, we need to make sure that our similarity term is invariant to geometric transformations (e.g., translation, rotation or scaling of photos, or part of the photos being cut and pasted onto other spam images, etc.). Moreover, since it is not uncommon that the same product will appear in different photos with different backgrounds, our measure should also be able to localize the

objects of interest even with background clutters. Both requirements imply that the global image features such as color histograms are not suitable in this case.

We adopt SIFT (Lowe, 2004), a local feature detector, to locate a number of feature points within the illustration regions, and use the percentage of matched features between two illustrations as their similarity measure. A class of local interest region descriptors are surveyed by Mikolajczyk & Schmid (2005) and SIFT is found to have the best performance among others. Given an input image, SIFT starts with detecting local extremes in a series of difference of Gaussian (DOG) functions over the scale space, with sub-pixel accuracy achieved by interpolating a local maximum with a 2D quadratic. For each feature location, one or more dominant orientations are determined, so that the features are invariant to image rotation. Finally, a descriptor for each feature point is formed by accumulating and bi-linearly interpolating local image gradients weighted by a Gaussian window, which provides certain degree of invariance to affine transformations. For a typical 400×300 image, SIFT is able to generate hundreds of feature points.

Once all picture illustrations in spam images are processed and a database of SIFT features generated, we can identify the number of matches between illustrations in any two spam images and determine their similarity. A match of a SIFT feature is identified as its nearest neighbor in the Euclidean space (Lowe, 2004). We adopt the ANN (Approximate Nearest Neighbor) package (Mount & Arya, 2006) for this task, which provides an efficient nearest neighbor search algorithm based on *kd*-tree.

The feature matching is performed on picture illustrations only. After feature matching, a similarity score is given to each pair of spam images based on the number of matches found between their picture illustrations. Specifically, we define the similarity score as

$$similarity(I, J)=matches(I, J)/min(Number\_of\_features(I), Number\_of\_features(J)) \quad (2)$$

which ranges from [0, 1] with ascending similarity. The intuition for using the size of the smaller feature set is that, if a part of the photo (containing the product) is cropped and pasted onto another spam image, the similarity between them will still likely to be high, because both the numerator and the denominator will decrease, so this measure is less biased for part-to-whole matching.

Figure 5 provides more outputs of our algorithm. We manually collect several product categories (see Figure 5(a)-(f)) and compute a similarity matrix within each category as well as the average similarity score among them. Figure 6 shows the average similarity scores of Figure 5(a)-(f) for an example of measuring unrelated spam images. The similarity matrix is generated by calculating the pair-wise similarities of images in the same category. Therefore this matrix is symmetric. Since similarities range from [0, 1], they can be easily visualized by converting them to gray-scale intensities as shown in Figure 6. The average similarity is the mean of all entries in the similarity matrix. Notice in Figure 5(f) that SIFT seldom produces false positives between different categories. Figure 7 is an example of calculating the similarity between two images according to Equation (2). The number of matches is 116 and the minimum number of features extracted for these two images is 331. Therefore, the similarity is approximately 0.35 (116/331).

Figure 5. Similarity matrices computed from individual categories (a)-(e) and a mixture of categories (f). (c) and (d) present some difficulties for SIFT because there are more modifications on the photos. However, the average score is still > 0.35. In (e), the watch in the last five spams is the same as the one in the middle from the first 12 spam images, but SIFT fails to find matches between the two groups.

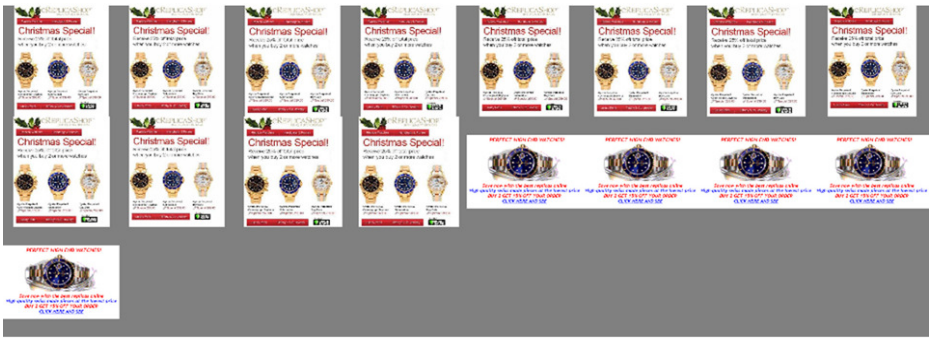
(a) watch(1)

(b) HoodioLife

(c) software

(d) website

(e) watch(2)



(f) spam photos from different categories

Figure 6. Similarity Matrix

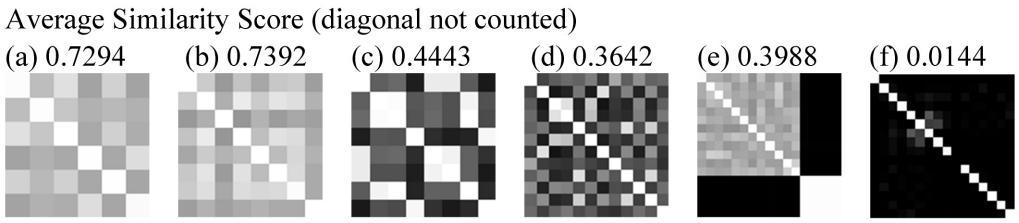
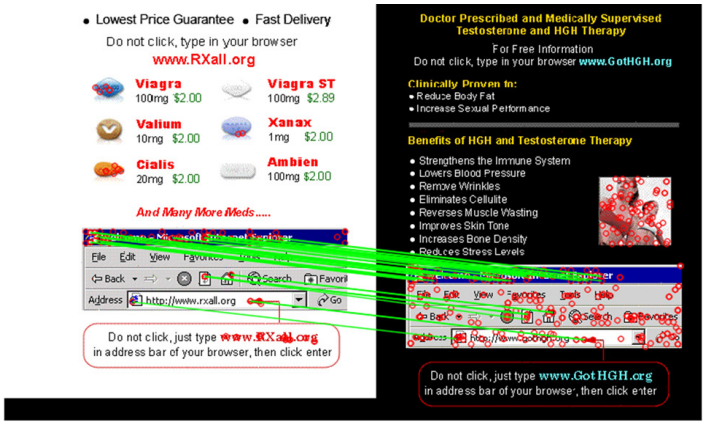


Figure 7. Two spam images that contain the same IE window but very different backgrounds. The image on the right contains another photo, which produces a lot more features, but the overall similarity is not reduced (similarity =116/331=0.35).



BACKGROUND TEXTURE ANALYSIS

When editing spam images, it is probably the easiest to change its background color to make it unique. Even created from the same template, the background colors (and sometimes even the foreground texts) may be different. Thus, color similarity cannot be treated as an important indication of common templates. Instead, we first convert the image background into grayscale. We further find that, although different in color, the background texture features of images created from the same template tend to have less variation. Therefore, in this paper, we analyzed the "homogeneity" and the "orientation" texture features of image backgrounds and found that with our currently colleted spam images, "orientation" textures can better distinguish among different templates than "homogeneity". In this study, we analyzed the "homogeneity" and the "orientation" texture features of the image background.

The homogeneity feature measures the closeness of the distribution of elements in the gray-level co-occurrence matrix to the diagonal of that matrix, where the gray-level co-occurrence matrix describes how often a pair of pixel intensity values is spatially correlated (Haralick, Shanmugam, & Dinstein, 1973). In this study, we create a series of gray-level co-occurrence matrices with various offset values (from -4 to 4). These offset values represent the window size used to examine the spatial relationship between pixel pairs.

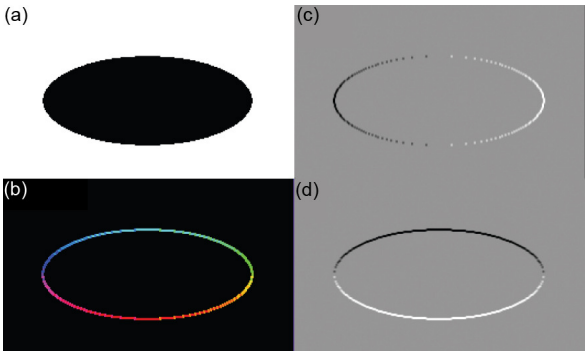
The orientation feature used in this study is an adapted version of directionality feature (Tamura, Mori, & Yamawaki, 1978), which measures the local direction of the edge in the

background textures by first applying the Prewitt edge operators, and then, computing the local orientation angle θ with the following formula (Burger & Burge, 2007).

$$\theta(u,v) = \tan^{-1}\left(\frac{\Delta_y(u,v)}{\Delta_x(u,v)}\right) \tag{3}$$

where (u, v) is the coordinates of an edge pixel; Δx and Δy are the filter results obtained from the corresponding Prewitt edge operators. Figure 8 illustrates how the Prewitt edge operators detect the edges and their local orientations.

Figure 8. Edge orientation detection: (a) the original image, (b) the edge orientations (represented as the hue change), (c) and (d) the edges detected by Prewitt edge operators X and Y



The obtained edge orientation values are then quantized into a 16-bin histogram H_{dir}. According to Tamura's paper, the directionality feature is the sum of second moments around each peak in H_{dir} from valley to valley. However, this measurement may cause problem since we may obtain the same directionality feature from two different H_{dir}. Hence, we adopt the normalized H_{dir} (divided by the total number of edge pixels) to represent the orientation feature of the background texture. The background texture similarity can be simply measured by the Euclidian distance between two texture feature vectors. According to our experiments, 'orientation' feature is significantly better than 'homogeneity' when we compare their distinguishing powers in classifying background textures.

Figures 9 and 10 show an example of spam image clustering based on the two texture features - homogeneity and orientation, respectively. In these two figures, the top-left image is selected as the cluster centroid and the top 20 nearest images are displayed from top to bottom and left to right in the descending order of their background texture similarity to the centroid. In Figure 9, only 7 out of the top 20 images have similar background texture as that of the centroid image, while in Figure 10 all top 20 images have similar background texture as the centroid, despite the disparity of background colors and texture scales.

Figure 9. Background texture similarity based on the homogeneity

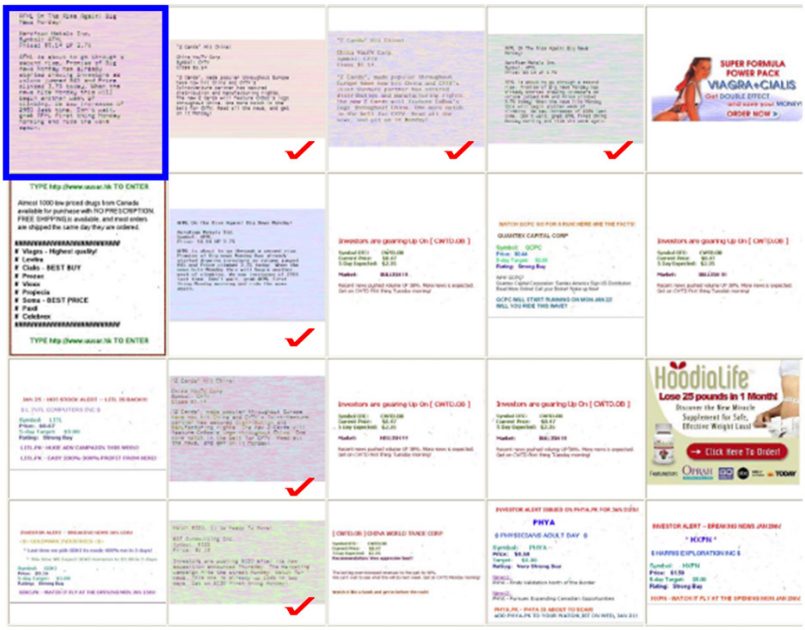
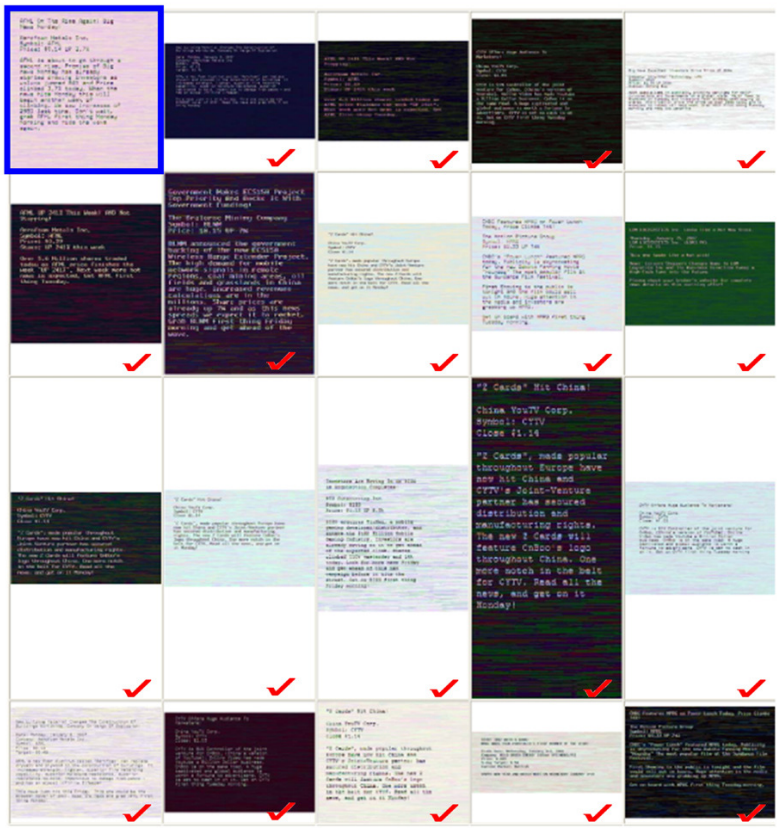


Figure 10. Background texture similarity based on the edge orientation



CLUSTERING

In the spam image clustering problem, we do not have a prior knowledge as to the number of spam clusters. Therefore, in order to approximate the number of clusters, a bottom-up agglomerative clustering method (Han & Kamber, 2000) is used to group spam images based on similar values of spam image features, including the text layout, SIFT features of picture illustrations, and background texture features. In the beginning, each spam image by itself is a single cluster. These initial clusters are at the leaf level of a hierarchical cluster tree. Then each nearest pair of clusters is merged together at each higher level of the tree. A non-leaf node represents a cluster formed through the merging of its two children nodes (clusters). The root of the tree is a cluster that contains all images. In measuring the distance between two images, we use the following formula:

$$d(I_i, I_j) = eucl(texture(I_i), texture(I_j)) + layout(I_i, I_j) + fgImage(I_i, I_j) \quad (4)$$

where the first term is the Euclidean distance of the "orientation" texture features of the two images I_i and I_j; the second term is the layout distance of two images; and the third term is the foreground picture illustration distance of the two. The first two terms can be easily obtained from texture analysis and foreground text layout analysis. As for the third term, as mentioned in Section 4, we compute the similarity matrix from the foreground picture illustration matching. The similarities are further converted to distances by deducting each entry in the similarity matrix from the maximum similarity. In cases where a pair of spam images both contain only texts but no foreground picture illustrations in either of them, their 3rd term distance is set to 0. When only one of the two images contains picture illustrations, their 3rd term distance is set to the maximum distance. This is to make sure that a pure text image is closer to another pure text image than to an image that contains foreground picture illustrations. Finally, all three terms are normalized by z-score (Larsen & Marx, 2000) before they are summed up to calculate the overall distance value between two images.

To estimate the approximate number of clusters, we need to cut the inconsistent links in the hierarchical tree. The inconsistent links are decided by the inconsistency coefficients of each link. The inconsistency coefficient characterizes each link in a cluster tree by comparing its length (distance) with the average length (distance) of other links to a certain depth of the hierarchy. The higher the value of this coefficient, the less similar the objects connected by the link. The cluster tree is then partitioned into clusters by setting a threshold on the inconsistency coefficient.

EXPERIMENTS

Spam Image Data Set

The spam images used in our experiments consists of those extracted from one month of emails manually identified as spam. We collect a high volume of spam through the use of "catch all" email addresses. A "catch all" configuration accepts mail for all possible addresses at a given domain. One common technique spammers use to "harvest" new target addresses is to send emails to randomly generated user IDs at well-known domains. Mail which does not "bounce" or reject is assumed by the spammer to have been delivered. Because a "catch all" address configuration accepts ALL mail, spammers treat all tested addresses as valid for this its domains. We test our algorithm on 1190 spam images. After clustering, there are 53 clusters in total.

Evaluation of Clustering Results

It is necessary to determine whether the resulting spam image clusters are meaningful in order to aid cybercrime investigation. Since we do not have the ground truth for the sources of the images, clusters were evaluated based on the visual characteristics (appearance) of these images at this point. Where the spam images from a single cluster demonstrate similar visual characteristics (e.g., text layout, picture illustrations, and/or background textures), a high confidence was placed upon the integrity of the cluster. These common visual characteristics may indicate the common source of those images from a single cluster.

The largest clusters are the 12th cluster (284 images), the 35th cluster (265 images), the 51st cluster (189), and the 47th cluster (133 images). Sample images of the four clusters are provided in Figure 11.

Figure 11. Sample images from the largest clusters

In Cluster 12, 256 out of 284 (i.e., 90.1%) images have background textures similar to the sample images in Figure 11. In addition, all the images in this cluster contain text areas only in the foreground. This cluster therefore is formed mainly because of the similarity of background textures and the text-only property. In Cluster 35, 252 out of 265 (95.1%) images are variations of the sample images shown in the Figure 11. When we trace back the cluster formation process, this cluster is formed mainly because of the layout similarity. Cluster 51 is comparatively not very satisfactory. It is composed of 1 major type (120 out of 189) represented by the third sample image. This cluster also has 4 minor types of images as shown by the other sample images. These minor types are mixed with pure text images (like the third sample image) because of the noise introduced by OCR text detection. When a block of texts are missed by OCR, these texts will be considered as foreground picture illustrations and compared with other true illustrations, resulting a less than maximum distance for the 3rd term in Equation (4). Although this cluster is not uniform since it combines one major cluster with three other small clusters, it can still provide a hint of common source at least for the spam images in the major cluster. The dominant features in Cluster 47 are background texture and text layout. Particularly, 125 out of 133 (94.0%) images in this cluster has "random dots/dashes" texture feature as shown in the sample images in Figure 11.

Figure 12. Sample images from Cluster 53

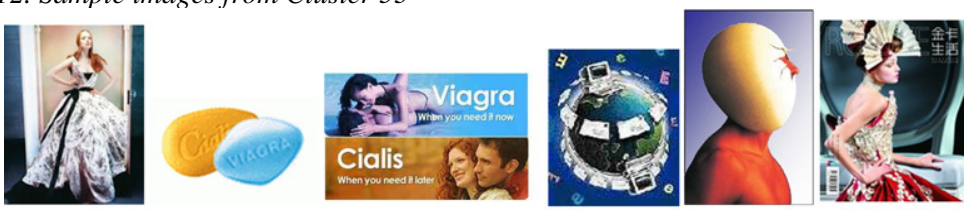


Figure 13. Sample images from Cluster 50

The rest of clusters are comparatively small, with the number of images ranging from 1 to 55. 34 clusters have less than 10 images in them. One kind of those clusters represents outliers. For example, as shown in Figure 12, there are in total 7 images in Cluster 53 which belong to 6 different types of images and all of them are pure images with no texts. The rare occurrence of these images makes it hard to track the originator and therefore can be ignored at this stage until more such data can be collected. The other kinds of small clusters are those which contain very similar foreground picture illustrations. For example, Cluster 50 has 8 images and all of them belong to one of the sample images shown in Figure 13.

The two types of website images illustrated in Figure 7 are mixed with text images in Cluster 52 (47 images) due to the noise caused by OCR as we mentioned earlier. However, with a deeper look into the cluster tree, both of these two types of website images are grouped in one cluster at an early stage of the tree construction and were merged into another cluster later.

CONCLUSIONS

This paper has proposed a new approach to advanced analysis of spam emails with a focus on the needs of law enforcement personnel. Using this approach, clusters of spam used for spreading messages to encourage the purchase of a product or service through image attachments can be readily identified. Furthermore, this approach can help to automate the process of visual validation of the spam clustering results, which are usually generated by analyzing the non-graphic information of spam messages such as email attributes (Chun et al., 2008) and textual email contents (Airold & Malin, 2004).

Given a spam image, the proposed approach first separates its foreground from the background. The foreground is further segmented into texts and picture illustrations. Foreground text layout is analyzed through a 4-step process – bounding box extraction, dilation, scaling, and similarity measuring. A feature matching method – SIFT is applied in the foreground picture illustration matching. For background analysis, we first perform the grayscale conversion and then analyze the texture features of the grayscale image background. Particularly, the "orientation" and "homogeneity" of background textures are extracted and compared. The "orientation" feature is finally chosen because of its better distinguishing power for our collected image data. Finally, we apply the agglomerative clustering method to group images into clusters. Our initial experiment showed promising results as significant clusters of emails were found which through the visual verification were shown to be tightly related, regardless of the variations in the image scale, background color and/or texture, or spatial placement of text and/or picture illustrations in the foreground. The result is not perfect as we are still exploring and improving our methods. But we believe it is a promising research area that worth further pursuit.

FUTURE WORK

Spam image mining is a new area and there is relatively little related work. To our best knowledge, our work is among the first to address spam image mining from the perspective of spam cluster identification. We attempted to address various issues in spam image clustering in this paper. This is an area that has a lot more to explore further.

The next stage of the research is to introduce more image features into analysis, especially color features. Although color features are not critical for background classification, it may improve the matching accuracy of the foreground images. We also plan to incorporate the text clue extracted by OCR into the clustering process. Another direction we will explore is the relative spatial relationship among the foreground objects including texts and foreground illustrations. Their relative positions may also be an important indicator of the editing style of spam images. The next issue is feature selection and information fusion from multi-modalities. When conducting this study, we find that one feature may be effective in differentiating a certain group of images, while it may fail on another group of images. Therefore, our long term goal is to build a feature selection model that can automatically select features and/or update the way of combining multiple features in calculating the distance value. This model shall have the ability to distinguish a large variety of spam images and adjust itself when new spam images are collected.

REFERENCES

Airold, E. & Malin, B. (2004). *Scamslam: An architecture for learning the criminal relations behind scam spam*. Carnegie Mellon University, School of Computer Science, Pittsburg Technical Report CMU-ISRI-04-121.

Burger, W. & Burge, M. J. (2007). *Digital image processing: An algorithm introduction using java*, 1st ed. Springer, New York.

Byun, B., Lee, C.-H., Webb, S., & Pu, C. (2007). A discriminative classifier learning approach to image modeling and spam image identification. *4th Conference on Email and Anti-Spam*.

Carreras, X. & Mrquez, L. (2001). Boosting trees for anti-spam email filtering. *International Conference on Recent Advances in Natural Language Processing*, (pp. 58-64).

Chun, W., Sprague, A., Warner, G., & Skjellum, A. (2008). Mining spam email to identify common origins for forensic application. *23rd Annual ACM Symposium on Applied Computing*, (pp. 1433-1437).

Clark, J., Koprinska, I., & Poon, J. (2003). A neural network based approach to automated e-mail classification. *IEEE/WIC International Conference on Web Intelligence*, (pp. 702-705).

Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5), 1048-1054.

Han, J. & Kamber, M. (2000). *Data mining: Concepts and techniques*. Morgan Kaufmann.

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transaction on Systems, Man, and Cybernetics*, 3, 610-621.

Larsen, R. J. & Marx, M. L. (2000). *An introduction to mathematical statistics and its applications*, 3rd ed. Prentice Hall.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 64(2), 91-110.

Mehta, B., Nangia, S., Gupta, M., & Nejdl, W. (2008). Detecting image-based email spam using visual features and near duplicate detection. *17th International World Wide Web Conference*, (pp. 497-506).

Mikolajczyk, K. & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1615-1630.

Mount, D. M. & Arya, S. (2006). ANN: A library for approximate nearest neighbor searching. *http://www.cs.umd.edu/~mount/ANN/*.

Rivest, R. (1992). The md5 message-digest algorithm. RFC 1321.

Sanpakdee, U., Walairacht, A., & Walairacht, S. (2006). Adaptive spam mail filtering using genetic algorithm. *8th International Conference on Advanced Communication Technology*, (pp. 441-445).

Tamura, H., Mori, S., & Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transaction on Systems, Man, and Cybernetics*, 8, 460-472.

Wu, C.-T., Cheng, K.-T., Zhu, Q., & Wu, Y.-L. (2005). Using visual features for anti-spam filtering. *IEEE International Conference on Image Processing*, (pp. III-509-512).

www.cnn.com/2007/TECH/11/29/fbi.botnets.