IMAGE SPAM HUNTER

Yan Gao, Ming Yang, Xiaonan Zhao, Bryan Pardo, Ying Wu, Thrasyvoulos N. Pappas, Alok Choudhary

EECS Dept., Northwestern Univ.
2145 Sheridan Rd., Evanston, IL 60208
{y-gao2,m-yang4,xiaonan-zhao,pardo,yingwu,t-pappas,anc123@northwestern.edu}

ABSTRACT

Spammers are constantly creating sophisticated new weapons in their arms race with anti-spam technology, the latest of which is image-based spam. The newest image-based spam uses simple image processing technologies to vary the content of individual messages, *e.g.* by changing foreground colors, backgrounds, font types, or even rotating and adding artifacts to the images. Thus, they pose great challenges to conventional spam filters. In this paper, we propose a system using a probabilistic boosting tree to determine whether an incoming image is a spam or not based on global image features, *i.e.* color and gradient orientation histograms. The system identifies spam without the need for OCR and is robust in the face of the kinds of variation found in current spam images. Evaluation results show the system correctly classifies 90% of spam images while mislabeling only 0.86% of non-spam images as spam.

***Index Terms*—** Image spam, probabilistic boosting tree

1. INTRODUCTION

Spam is e-mail that is both unsolicited by the recipient and sent in substantively identical form to many recipients. As of December 2006, Infoweek reported that 94% of all electronic mail is now spam, making spam filtering very important for the continued utility of electronic mail.

Currently, spam is mainly dealt with on the receiving end by automated spam filter programs that attempt to classify each message as either"spam" (undesired mass email) or "ham" (email the user would like to receive). Current spam filtering programs treat spam detection as a text classification problem, utilizing machine-learning algorithms such as neural networks and naive Bayesian classifiers to learn spam characteristics. Among these, Bayesian-based approaches [1, 2] have achieved outstanding accuracy and have been widely used. These spam filters can adapt their classification engines as spam texts vary with time, learning from corrections provided by end users.

Recently, spammers have begun evading filters by encoding the spam messages as images. Typically the image contains a screen shot offering the same types of information advertised in traditional text-based spam, though it may contain some pictorials. This image is typically attached to or embedded in a message whose text contains randomly generated words, excerpts from famous literature or even excerpts from private non-commercial emails. This type of image spam accounted for 65% of all global spam by the end of 2006, compared with just 30% by the start of 2006 [3]. This presents a problem for current spam filters, as text-based spam filtering does not work on such image-based spam.

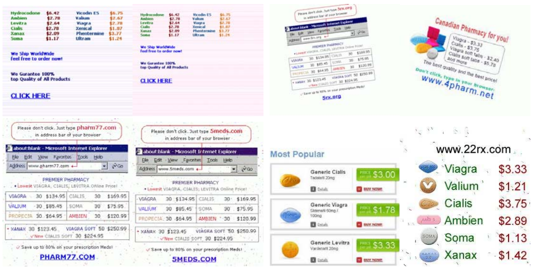

Hydrocodone $6.42 Vicodin ES $6.75
Ambien $2.78 Valium $2.67
Levitra $2.64 Viagra $2.78
Cialis $2.78 Xenical $1.87
Xanax $2.09 Phentermine $3.77
Soma $1.17 Ultram $1.24
We Ship WorldWide
feel free to order now!
We Gurantee 100%
top Quality of All Products
CLICK HERE
Canadian Pharmacy for you!
Viagra - $3.33
Cialis - $3.75
Viagra soft tabs - $2.40
Cialis soft tabs - $5.78
and more
The best quality and the best price!
Don't click, type in your browser:
www.4pharm.net
5rx.org
Please don't click. Just type pharm77.com in address bar of your browser
about:blank - Microsoft Internet Explorer
PREMIER PHARMACY
VIAGRA 30 $134.95 CIALIS 30 $169.95
VALIUM 30 $85.45 SOMA 30 $75.95
PROPECIA 30 $64.95 AMBIEN 30 $120.99
XANAX 30 $123.45 VIAGRA SOFT 50 $250.99
New CIALIS SOFT 30 $224.95
Save up to 80% on your prescription Meds!
PHARM77.COM
Please don't click. Just type 5meds.com in address bar of your browser
5MEDS.COM
Most Popular
Generic Cialis $3.00
Generic Viagra $1.78
Generic Levitra $3.33
BUY NOW!
www.22rx.com
Viagra $3.33
Valium $1.21
Cialis $3.75
Ambien $2.89
Soma $1.13
Xanax $1.42

Fig. 1. Sample spam images: image size changes and rotation (1st row), artifacts in the background and images with icons (2nd row).

There are several organizations and companies working on ways to filter image-based spam. SpamAssassin (SA) [4] is a widely-deployed filter program that uses Optical Character Recognition (OCR) software to pull words out of the images and then uses the traditional text-based methods to filter spams. Since spammers use the same obfuscation techniques they have long used in text-based spams, *e.g.* misspelling words, fuzzy matching was added to the filter. Spammers have responded by including a light background of random artifacts or rotating the image slightly. These practices do not affect the readability to humans, but do greatly affect the quality of the OCR output. This greatly increases the difficulty of spam filtering that relies on OCR.

To foil spam filtering based on matching existing images to previously encountered spam images with image comparison techniques, spammers randomly tile images and include varied borders or backgrounds, randomly varied spacing or margins, and add specks to the image. The consequence is a huge quantity of image-based spams that contain random patterns with few exact repetitions. Sample spam images are shown in Fig. 1 to show the diversity of spam images.

Given the recent upsurge in image-based spam, we are interested in developing a method to filter spam based on image content, rather than text content. This approach should be robust in the face of the kinds of variations introduced to foil OCR-based spam filtering (random patterns of dots and lines, image rotation). The approach should also be robust to random re-tiling of images and variations in borders, margins and fonts. Thus, it should let us identify groupings of similar but not-identical images [5]. Finally, this approach should be trainable from real-world data without the need for hand-coded heuristics.

It is an important requirement that spam detection system can tolerate some false negatives but cannot afford false positives (desired photos and image attachments falsely classified as spam). Currently most email service providers (such as hotmail and gmail) provide the customers a *junk* button, which not only helps spam filter to collect spams, but also lets customers identify useless spam emails incorrectly classified as good mail (false negatives). We wish to design a system that is biased towards minimization of false positives at the expense of allowing a few spam images through.

In this paper, we propose a learning-based prototype system *Image Spam Hunter*, as shown in Fig 2, to differentiate spam images from normal image attachments. We first cluster the collected disordered spam images into groups based on image similarity measurement on global color and gradient orientation histograms [6]. The training dataset is chosen from the clustered groups. We then build a probabilistic boosting tree (PBT) [7] based on the training dataset to distinguish image spams from good emails with image attachments. Image Spam Hunter learns to distinguish spam from ham images without need for performing OCR on the image, and is robust in the face of the kinds of random variation that exist in current spam images. The proposed method achieves 0.86% false positive rates versus 89.44% true positive rates in 5-fold cross-validation.

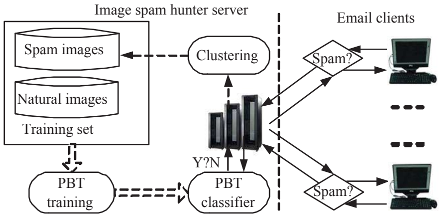


Fig. 2. Prototype system diagram.

2. OUR APPROACH

Though OCR may be the ultimate solution to the image spam problem, current text OCR techniques are quite computationally demanding and vulnerable to image artifacts. Intuitively, for most of the cases, people don't need to recognize the texts in the images to determine if they are more likely to be spams. Since the spam images are artificially generated, we expect their image texture statistics are distinguishable from the kinds of images typically included as attachments to personal emails. Such normal images are typically photographs of natural scenes including items such as sea, buildings, and humans. Therefore, we propose to employ a probabilistic boosting tree to differentiate spam images from normal images based on efficient global image statistics, *i.e.* color and gradient orientation histograms.

2.1. Image feature extraction

We consider two cues, color and gradient orientation histograms, as the features for classification. The observation is that most of spam images are converted from text spams, although they may contain some icons and artifacts. Thus, the range of color components in a typical spam is quite limited compared with a natural scene. As shown in Fig. 3, the color histograms of natural scenes tend to be continuous, while the color histograms of artificial spam images tend to have some isolated peaks. Another observation is that the distribution of gradient orientation may reveal the characteristics of texts. Fig. 4 illustrates the comparison of 1D histograms of gradient orientation of spam and natural images. The distributions of gradient orientation for natural images appear more uniform and noisy than those of spam images. Gradient orientation histograms are particular effective to deal with gray-level images.

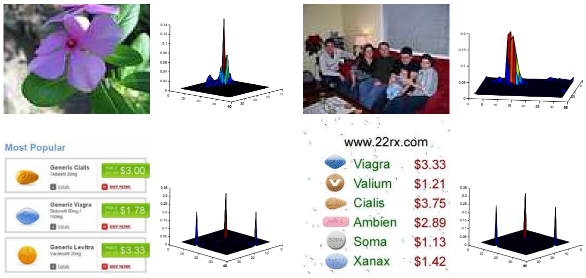


Fig. 3. Color histograms comparison between natural images and spam images in 32 × 32 2D normalized RG plane.

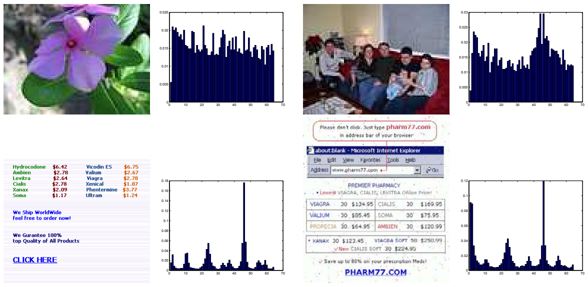


Fig. 4. Gradient orientation histograms comparison between natural images and spam images.

Specifically, we extract two N dimensional feature vectors, color vector $\mathbf{x^c} = \{\mathbf{x^c_1}, \cdots, \mathbf{x^c_N}\}$ and gradient orientation histogram vector $\mathbf{x^g} = \{\mathbf{x^g_1}, \cdots, \mathbf{x^g_N}\}$ for each image. Each color can be represented by two independent components, so we build 2D color histograms in a particular color space (normalized RG space in our experiments). Since we only care about the shape or color distribution rather than the exact meaning of color bins, we sort the bins in descending order and only keep the top N bins as the feature vector $\mathbf{x^c}$.

This approach also balances the need for high resolution in the color domain (to avoid quantizing similar colors to the same bin) against the need for efficient training and testing in reasonable high dimensional space. In our experiments, we calculate $32 \times 32 = 1024$ 2D color histogram and keep the top $N = 32, 64, 128$ unique bins. To extract the gradient orientation histogram $\mathbf{x^g}$, the image gradient for each pixel is calculated with a Sobel operator. If the gradient magnitude is larger then a threshold $t_m = 50$, we quantize its orientation angle $0^o - 360^o$ to one of N bins.

2.2. Training set generation

We collected two sets of images to train our filter: normal images and spam images. Since we anticipate normal images will typically be the kinds of images found in image-sharing social networking sites, we collected the normal images by downloading 830 randomly selected images from Flickr.com. We collected 928 spam images from real spam emails as the spam sample set. These images were drawn from the image spams received by the authors in the last 6 months.

Treating the collected spam images appropriately is not trivial. There are two important concerns. Firstly, it is very likely that a lot of repetitive spam images are reported by customers in a short period. If the training set includes all of them or is generated by random selection, certain type of spam images may dominate the training set. Secondly, different people may have different spam definitions, *e.g.*, fur ads may appear disgusting to pet lovers but attractive to others, images that are inconsistently reported as spam should not be used as training samples. Therefore, we need to generate the training set for learning through a rough clustering rather than pure random selection.

There are many clustering algorithms, which can be used in our prototype system. In this paper we cluster the spam images using agglomerative hierarchical clustering approach, which automatically stops at a certain threshold for intra-class distance. Clusters with less than a certain number of the spam images are excluded from training, and thus the training samples are always selected from the large clusters. We adapt the clustering results with the new incoming images by comparing the ratio of the distance to the nearest and the second nearest cluster. If this ratio is larger than the threshold, *i.e.* 0.9, the new image will be in a new cluster, and vice versa. The thresholds above should be experientially adjusted in terms of the different email service sizes.

We perform the hierarchical clustering above by using the combination of color and gradient orientation histogram feature vectors to all the spam images in our dataset and keep the largest k groups($k = 6$ empirically selected). We ensure the training set represent various types of spam by selecting the same number of images from each group. For example, for 5-fold cross-validation, training set is made up of about 4/5 samples from each group. Note we do not cluster the ham images.

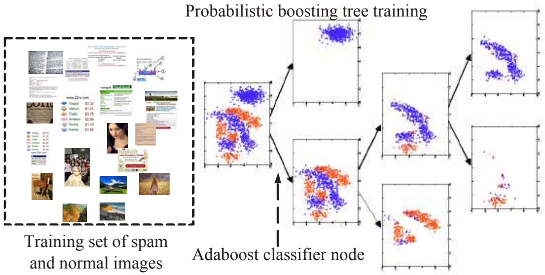

Probabilistic boosting tree training
Training set of spam and normal images
Adaboost classifier node

Fig. 5. Illustration of PBT structure.

2.3. PBT classification

Image similarity measurement is an active and open research topic that is generally very difficult. In this paper, we aim to merely distinguish a specific group of images, *i.e.* the spam images, from normal images by supervised learning. Thus, we collect training image samples I_i and represent them with feature vectors with labels $(\mathbf{x}_i^c, \mathbf{x}_i^g, y_i)$, where $y_i = +1$ indicates spam image and $y_i = -1$ for normal image.

We employ a probabilistic boosting tree [7] (PBT) method to classify the spam and natural images. Essentially, a PBT is a decision tree trained with positive (spam images) and negative (normal images) samples, where each node in the tree is an Adaboost classifier. The Adaboost algorithm learns a strong classifier $H(\mathbf{x})$ by combining a set of weak classifiers $h_t(\mathbf{x})$ as $H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$. Denote the probabilities computed by each learned Adaboost classifier as

$$p(+1|\mathbf{x}) = \frac{exp\{2H(\mathbf{x})\}}{1 + exp\{2H(\mathbf{x})\}}, p(-1|\mathbf{x}) = \frac{exp\{-2H(\mathbf{x})\}}{1 + exp\{-2H(\mathbf{x})\}}.$$

At each node, the training samples are divided into two overlapped sets $S_{left} = \{(\mathbf{x}_i, y_i)|p(+1|\mathbf{x}_i) > 0.5-\epsilon\}$ and $S_{right} = \{(\mathbf{x}_i, y_i)|p(-1|\mathbf{x}_i) > 0.5 - \epsilon\}$, then these two sets are passed to left and right sub-trees to further train Adaboost classifiers. The classification result for a feature $\mathbf{x}$ combines the probability at every node in a probabilistic way,

$$\begin{aligned} p(y|\mathbf{x}) &= \sum_{l_1} p(y|l_1, \mathbf{x}) p(l_1|\mathbf{x}) \qquad (1) \\ &= \sum_{l_1, l_2} p(y|l_2, l_1, \mathbf{x}) p(l_2|l1, \mathbf{x}) p(l_1|\mathbf{x}) \\ &= \sum_{l_1, \cdots, l_n} p(y|l_n, \cdots, l_1, \mathbf{x}), \cdots, p(l_2|l1, \mathbf{x}) p(l_1|\mathbf{x}), \end{aligned}$$

where the tree level l_i is an augmented variable and $p(l_i|\mathbf{x})$ denotes classification probability of the Adaboost classifier for this testing feature $\mathbf{x}$ at level l_i. An illustration of the hierarchical PBT is shown in Fig. 5.

Due to the independency of the color and gradient orientation histograms, we train two PBTs for $\mathbf{x^c}$ and $\mathbf{x^g}$ respectively. Our experiments show that the false positives of two PBTs are less than one combined PBT. The classification probabilities calculated by Eq.1 are denoted as $p(\pm1|\mathbf{x}^c)$ and $p(\pm1|\mathbf{x}^g)$ for a testing image. The image is marked as spam if

N	Accuracy	FP Rate	TP Rate
32	0.5631	0.0136	0.1944
64	0.9494	0.0420	0.9417
128	0.9471	0.0469	0.9426

Table 1. Comparison of 5-fold cross-validation performance of different D dimensional vectors ($\delta = 0$).

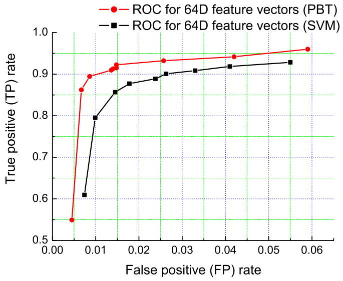


Fig. 6. ROC curves for 64D feature vectors using PBT and SVM classifiers, respectively.

and only if both PBTs make the same decision, so as to avoid false positive as much as possible,

$$y = \begin{cases} +1 & p(+1|\mathbf{x}^c) > 0.5 + \delta \text{ AND } p(+1|\mathbf{x}^g) > 0.5 + \delta \\ -1 & \text{otherwise.} \end{cases} \quad (2)$$

where δ is a parameter to adjust the performance.

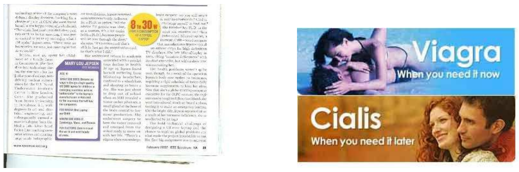
We employ normalized RG space in color histogram calculation, which is insensitive to lightings. The color histograms are sorted in decreasing order and truncated to keep top N bins. In our PBT implementation, we employ the Gentle Adaboost classifier in the OpenCV library [8] at each node which consists of 100 decision stumps, *i.e.* 1-level decision trees, as weak classifiers. The ϵ is set to 0.1 to spilt the tree.

3. EXPERIMENT

We test the spam detection by 5-fold cross-validation on the aforementioned database. There is no overlap of the training and testing sets. The performance is measured with the average false positive (FP) rate, *i.e.* the misclassification rate of normal images, true positive (TP) rate, *i.e.* the detection rate of spam images, and the overall average accuracy on both spam and normal image sets.

By testing different D values as in Tab. 1, we find $N = 64$ is sufficient to detect the spams in our current sample set. Classification accuracies over both training spam and normal images are listed in the table as well.

Fig. 6 displays the ROC curve of the PBT classifier applied to N=64 feature vectors, as well as the ROC curve of a support vector machine (SVM) [8] with radial-basis kernel applied to the same features. Each point in ROC curve of the PBT classifier corresponds to a different δ value. The SVM classifier is tested as the baseline for comparison, because it has been widely demonstrated that SVM classifiers may achieve better performance than other types of classifiers such as naive Bayesian classifier and neural network. From Fig. 6, we can easily observe that the PBT demonstrates significant performance gain over the SVM for this task. It achieves 89.44% detection rate at the FP rate of 0.86%, while SVM only achieves approximately 80% detection rate at the same FP rate, since the PBT tries to solve this extremely hard classification problem gradually. This preliminary result seems quite positive and acceptable for real email systems. Our approach tests one image within 0.4s on average on a Pentium 3G desktop .



(a) False positive (b) False negative

Fig. 7. Sample false positive and false negative.

Some incorrectly classified images are shown in Fig. 7. We found that they fall into two typical categories: scanned documents with icons, and spam images with a large percentage of the image covered by real life photos. These can easily be classified mistakenly even by humans, if the text content is not taken into account. Correctly classifying these images must rely on content analysis methods (such as OCR), which will largely increase the computational complexity.

4. CONCLUSION AND FUTURE WORK

In this paper, we proposed a prototype system to detect the spam images in email. The proposed method extracts efficient global image features to train an advanced binary classifier to distinguish the spam images, which achieves promising preliminary results on our sample database. It is still possible to fool our approach, given the gap between the global image features and the semantic meaning of the email content. Therefore, our future work will include some high level human vision models and more sophisticated analysis methods, and make the system accommodate for online learning.

5. REFERENCES

[1] J. Blosser and D. Josephsen, “Scalable centralized bayesian spam mitigation with bogofilter,” in *USENIX LISA*, 2004.

[2] Kang Li and Zhenyu Zhong, “Fast statistical spam filter by approximate classifications,” in *ACM SIGMETRICS*, St. Malo, France, June 2006, pp. 347 – 358.

[3] McAfee, “http://www.avertlabs.com/research/blog/?p=170,” .

[4] SpamAssassin, “http://spamassassin.apache.org/,” .

[5] Dong-Qing Zhang and Shih-Fu Chang, “Detecting image near-duplicate by stochastic attributed relational graph matching with learning,” in *ACM MM*, 2004, pp. 877 – 884.

[6] A. Maccato and R.J.P. deFigueiredo, “The image gradient histogram and associated orientation signatures,” in *ISCAS*, Seattle, WA, 1995, vol. 1, pp. 239– 242.

[7] Zhuowen Tu, “Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering,” in *ICCV*, 2005, vol. 2, pp. 1589–1596.

[8] Open Source Computer Vision Library, “http://www.intel.com/technology/computing/opencv/,” .