Proceedings of the 2007 IEEE
Workshop on Information Assurance
United States Military Academy, West Point, NY 20-22 June 2007

Identifying Image Spam based on Header and File Properties using C4.5 Decision Trees and Support Vector Machine Learning

Sven Krasser, Yuchun Tang, Jeremy Gould, Dmitri Alperovitch, Paul Judge

Abstract— **Image spam poses a great threat to email communications due to high volumes, bigger bandwidth requirements, and higher processing requirements for filtering. We present a feature extraction and classification framework that operates on features that can be extracted from image files in a very fast fashion. The features considered are thoroughly analyzed regarding their information gain. We present classification performance results for C4.5 decision tree and support vector machine classifiers. Lastly, we compare the performance that can be achieved using these fast features to a more complex image classifier operating on morphological features extracted from fully decoded images. The proposed classifier is able to detect a large amount of malicious images while being computationally inexpensive.**

I. Introduction

The volume of spam messages sent on a daily basis is alarming and poses a great threat to the utility of email communications. Since the first recorded spam email, which was sent on May 3, 1978 by a DEC Marketing representative, Gary Thuerk, to everyone at the time using ARPANET on the West Coast, the use of email for promotion and selling of both legitimate and fraudulent products has skyrocketed [1]. At this point, reading email without a spam filter in place is nearly impossible or requires both making email addresses difficult to guess for dictionary-based email harvest attack tools and placing rigorous restrictions on who to share the address with—defeating the idea of a ubiquitous, fast, and easy means of communication. By the end of 2005, approximately 70% of all email on the Internet was spam as determined by Secure Computing Research. Just one year later at the end of 2006, this value rose to about 90%.

A recent variant is image spam in which the actual spam message is moved into an image attached to the message. While at the end of 2005 one out of ten spam messages was an image spam, this number was up to one message in three at the end of 2006 based on data from Secure Computing Research.

Early approaches to combat this new type of spam have tried to extract text from these images using optical character recognition (OCR). This text can then be fed to the same textual classifiers as regular spam messages. Spammers are now specifically targeting these approaches by composing their images specifically in a way that makes it hard to use OCR on them. This includes low contrast, adding pixel noise, geometric structures in the background, sloped and wavy text, and using animation (animated GIFs). Ironically, many of these approaches are similar to what has been done in the scope of CAPTCHAs (an acronym short for *Completely Automated Public Turing test to tell Computers and Humans Apart*). Many CAPTCHAs are implemented as images that challenge humans to read the text they contain and enter it into a text box to proof that they are actually humans. A common use is the use of such techniques on websites to prevent programs from making automated requests.

A more promising path is to capture what makes spam images look like spam images mathematically and store it in a feature vector. Similarly to a human who does not need to read a spam image but can tell it most of the time apart from a good image (ham image) by simply glancing at it, such a classifier can detect new strains of spam without the need of signatures generated beforehand or the ability to extract text from an image. Secure Computing Research developed such a classifier based on about 100 different features capturing the composition of the image in the spatial and in the Fourier domain. However, such an image feature analysis is expensive when considering the large volumes of image spam messages currently circulating. Therefore, there is a need for a quick first sift to get rid of as many image spam messages as possible with as low computational effort as possible. One solution to this is to generate a first set of inexpensive features from the image data that are purely based on header information and image file properties to avoid the calculation of more expensive morphological features.

In this paper we present an algorithm and framework that is able to achieve this task and sift out a significant amount of spam images with minimal effort. The rest of this paper is organized as follows. Section II gives an overview about the history and evolution of spam leading up to image-based spam. Section III describes the features extracted to classify images. In Section IV we analyze these features with respect to their distribution and how much information they can contribute for classification. Section

S. Krasser, Y. C. Tang, J. Gould, D. Alperovitch, and P. Judge are with Secure Computing Corporation, 4800 North Point Parkway, Suite 400, Alpharetta, Georgia 30022.

1-4244-1304-4/07/$25.00 ©2007 IEEE

V outlines how these features can be effectively classified using C4.5 decision trees and support vector machines. The paper is concluded by Section VI.

II. History and Background

To better understand the current techniques and technologies spammers are employing in this never-ending battle to get their message into end-user inboxes, it is useful to look at the history and evolution of this fight.

A. The first Spam

In 1978 Gary Thuerk had to manually type in all the email addresses of his several hundred recipients into his email program to send out a single copy of the marketing promotion. He did not randomize the text or attempt to hide his identity—at the time spam filters had been unheard of, and canned meat was the only meaning that the word "spam" had described.

His followers in the mid-90s quickly realized that with a spread of the Internet, their potential customer audience had enlarged to millions and then quickly to tens of millions and their email clients and address books could no longer scale. Instead, they had to write custom email sending software to run on powerful rented servers with fast bandwidth connections to reach an audience that wide. Access to powerful servers and lots of bandwidth to use to send out millions of spam each day started to require substantial monthly investments on the part of the spammers.

B. Blacklists

In the late 1990s, as spam volumes had started to increase to noticeable levels and spam had started to become a nuisance, if not yet a security threat, security vendors and volunteers begun to look at solutions to identify and stop these unwanted messages. These techniques included identifying and blacklisting IPs of servers that spammers had employed to send out the emails, as well as writing some rudimentary rules and signatures to identify the common text used in the spam messages.

C. Botnets

The returns on the investments in servers and bandwidth were diminishing with each day as the IPs of these servers had been discovered and blacklisted, dramatically reducing deliverability. Often this also resulted in termination of the contracts spammers had with their service providers and the potential discovery of the identities of the spammers. That lack of anonymity started to become a major liability resulting in lawsuits and potential for criminal prosecution as countries around the world began to criminalize acts of sending unsolicited email.

It was then that some enterprising spammers had first realized that they can instead utilize potentially millions of machines around the world at almost no cost to them with near absolute anonymity. Some had partnered with virus and exploit writers to get access to the machines, known as zombies, that were being infected by their malware and install specialized SOCKS and SMTP proxy software on those machines to relay connections from the servers of the spammer through the zombies. This ensured that they had access to potentially millions of zombie IP addresses.

Identifying these in real-time would become a new challenge for the anti-spam industry, and tracking spammers from that point on to bring them to justice via the criminal or civil legal systems would become an extremely difficult international undertaking.

The zombie machines are often connected to the Internet via an always-on broadband connection and, in aggregate, their combined bandwidth can far exceed the bandwidth of dedicated servers spammers tended to use in the past. To further increase the speed and volume of their mailings, they have mostly abandoned the proxy and approach and today use fully automated and specialized mail server software running on millions of zombie computers worldwide to send out billions of emails each day.

This software can download (typically from a centralized Web server) a template of the email to send out and a list of recipient email addresses that can contain hundreds of millions of entries. A template can contain special sections that are to be replaced with random text with each mailing and the format in which to create RFC-822 email headers. The format of the headers typically emulates the format and algorithms used by popular client email software such as Microsoft Outlook or Outlook Express. The template can be changed at any point on the server to allow for a new spam run or modify the URLs in the email, which had become known and blacklisted by anti-spam solution providers with new yet-unused domain names. The spam software uses this template to send out tens of thousands of messages each hour from a single bot machine. Each email is unique due to the use of randomizations and can often pass undetected through signature-based anti-spam classifiers.

D. Image Spam

However, with increased effectiveness of text-based anti-spam classification engines, most notably Bayesian filters, in 2006 spammers once again raised the stakes and deployed new software on the zombies to convert the message template into an image attachment. By moving their message to an image, spammers try to avoid exposing usable tokens (words) to such textual classifiers. In addition, due to image compression even slight changes in an image, such as the introduction of pixel noise or randomization of the color palette, can have great ramifications on its binary representation making it infeasible to block images based on simple binary string signatures.

E. Outlook

At the current stage of this battle, few holistic and effective anti-spam approaches exist to combat this latest innovation. We believe the method and algorithm we propose in this paper can serve as a key tool that can be used to bring us back to the times when over 99% of the spam can be prevented from reaching an end-user's inbox.

III. Feature Extraction

We consider four basic features that can quickly be derived from an image at an extremely low computational cost. These are the width and the height denoted in the header of the image file, the image file type, and the file size. Based on these raw features, we generate a small 9 dimensional feature vector as shown in Table I. The image file type features (f_4, f_5, and f_6) are binary features that are set to 1 if the file is of the specified type and to 0 otherwise. In this research, we focus on the three dominant file formats commonly seen in email, which are the Graphics Interchange Format (GIF), the Joint Photographic Experts Group (JPEG) format, and the Portable Network Graphics (PNG) format.

A general idea of the image dimensions (i.e. width and height) can be gathered by parsing the image headers of the GIF, JPEG, or PNG files using a minimal parse. This is very fast since it doesn't decompress or decode any actual image data. Unfortunately we only get a general idea because obtaining the actual dimensions can be somewhat trickier and more time consuming in most cases.

In the case of GIF files (the current *de facto* standard for image spam) the presence of virtual frames, which can be either larger or smaller than the actual image width, is an issue that can only be detected while decoding the image data. Other embedded information such as alpha channel and multiframe images can require a full parse of the image data to detect.

Also, corrupted images can pose a problem. For corrupted images most typically some of the lines near the bottom of the image do not decode properly and no further image data can be decoded after that point. This is an issue for PNG and JPEG images as well, and seems to be one of the spammers favorite tricks. Reverse engineering analysis performed by Secure Computing of versions of spambot software responsible for generation and randomization of the images used in much of the image spam has uncovered memory leaks and other bugs in the image generation code that we speculate is occasionally introducing this corruption. As such, presence of corruption in the image currently happens to be a very good discriminator of spam and ham but if these bugs will eventually get discovered and addressed by the software authors, the feature may become less useful.

TABLE I
Image features.

#	Description
f_1	Image width denoted in header
f_2	Image height denoted in header
f_3	Aspect ratio: f_1/f_2
f_4	Binary: GIF image
f_5	Binary: JPEG image
f_6	Binary: PNG image
f_7	File size
f_8	Image area: $f_1 \cdot f_2$
f_9	Compression: f_8/f_7

IV. Feature Analysis

To evaluate the amount of information that can be gained from these features, we define the signal to noise ratio (S2N) as the distance of the arithmetic means of the spam and non-spam (ham) classes divided by the sum of the corresponding standard deviations (similarly to [2]):

$$S2N = |\frac{\mu_{\text{spam}} - \mu_{\text{ham}}}{\sigma_{\text{spam}} + \sigma_{\text{ham}}}|.$$

The results of this analysis are presented in Table II. Note that the means of the binary features reflect the percentages of images in the respective formats. An overwhelming amount of image spam uses GIF images (96.8 % as indicated in Table II). There are only few image spams using the JPEG format, even though the percentage of such spam images is increasing according to our data. The same data when only considering one image format at a time is presented in Table III for GIF, Table IV for JPEG, and Table V for PNG. Most legitimate images in emails (ham images) are JPEG images according to our sample corpus. We attribute this to photos shared, which are most commonly stored in the JPEG format due to its high compression ratio for photographic images. GIF only offer 256 colors, so that it is not a popular format for this kind of image. [1]

Feature f_9 is the most informative feature beyond the binary image format features. This feature captures the amount of compression achieved by calculating the ratio of pixels in an image to the actual file size. The higher this number, the better is the compression of the image (more pixels are stored per byte). We neglect the fact here that GIFs can contain multiple frames (animated GIFs). However, this information is not available in the header information and would require a full parse of the image. Figure 1 shows the probability distribution for ham and

[1] More precisely, GIF allows a color palette of 256 colors for each virtual frame inside a single file. These palettes can be different for each virtual frame, so that the final image rendered from a file can have more than 256 unique colors. However, this technique is rarely used.

TABLE II
FEATURE QUALITY.

#	S2N	μ_{spam}	σ_{spam}	μ_{ham}	σ_{ham}
f_1	0.007	516.100	175.940	508.893	830.512
f_2	0.045	354.951	128.957	396.522	802.890
f_3	0.024	1.770	4.522	18.083	671.160
f_4	1.032	0.968	0.176	0.309	0.462
f_5	0.992	0.029	0.168	0.664	0.472
f_6	0.110	0.003	0.054	0.027	0.161
f_7	0.240	15434.405	13470.418	150135.290	547899.901
f_8	0.171	193329.655	106852.331	384334.607	1013120.941
f_9	0.660	16.694	10.373	4.897	7.491

TABLE III
FEATURE QUALITY (GIF ONLY).

#	S2N	μ_{spam}	σ_{spam}	μ_{ham}	σ_{ham}
f_1	0.188	519.213	176.381	257.070	1216.534
f_2	0.143	356.251	128.666	165.079	1208.725
f_3	0.043	1.763	4.583	53.793	1206.121
f_7	0.100	15269.592	13459.129	29347.112	127587.524
f_8	0.767	195339.565	107180.158	42098.934	92658.873
f_9	0.524	16.974	10.363	5.003	12.503

TABLE IV
FEATURE QUALITY (JPEG ONLY).

#	S2N	μ_{spam}	σ_{spam}	μ_{ham}	σ_{ham}
f_1	0.289	422.083	133.165	618.413	546.647
f_2	0.308	305.491	129.184	496.662	491.595
f_3	0.004	2.050	2.005	2.123	14.980
f_7	0.272	21601.046	12787.328	203686.373	655880.879
f_8	0.323	127524.731	71339.823	539062.478	1202866.954
f_9	0.265	6.704	3.932	4.823	3.155

TABLE V
FEATURE QUALITY (PNG ONLY).

#	S2N	μ_{spam}	σ_{spam}	μ_{ham}	σ_{ham}
f_1	0.509	422.727	106.532	701.057	440.488
f_2	0.255	415.818	110.554	586.075	557.189
f_3	0.303	1.060	0.329	1.582	1.392
f_7	0.625	8708.182	5418.077	216762.736	327640.927
f_8	0.451	183087.273	81826.020	497976.736	616840.791
f_9	1.509	23.594	5.361	5.534	6.605

spam images for this feature (cut off at 30). While the fact that different image formats compress differently well and that most image spam seen on the Internet today is based on GIF images plays into the quality of this feature, it is not the determining factor. Looking at tables III to V, we can see that even among images of the same format there is a noticeable difference in the compression between ham and spam images. In Figure 2, we present the probability density when only considering GIF images. The separation is similar to the one observed in the overall data set. The

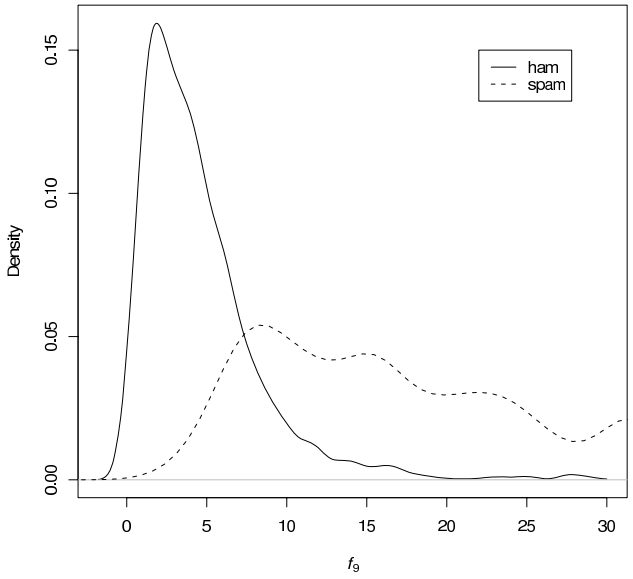

Fig. 1. Probability density for f_9.

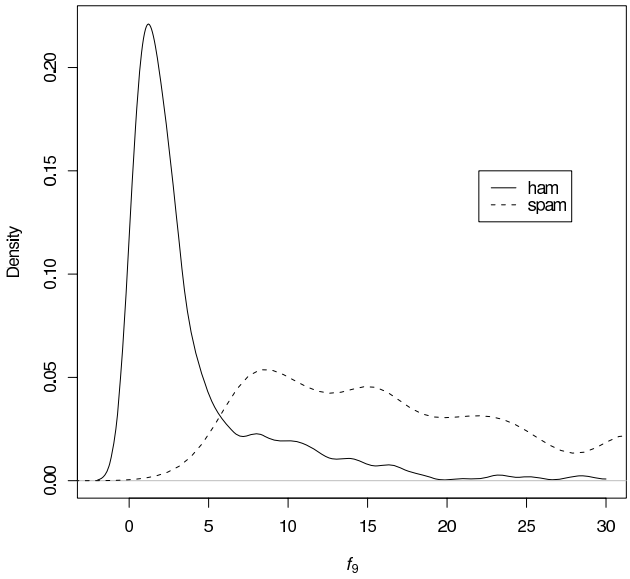

Fig. 2. Probability density for f_9 for GIF images only.

S2N value for f_9 for PNG images is the highest, which is partially due to the very small sample size in the corpus used.

In Figure 3, we show a contour plot of the two-dimensional probability distribution in the subspace spanned by f_9 and f_1. In this plot it can be observed that there are distinct clusters in which ham and spam images fall.

V. Classification

In one of our previous works, a classification modeling study demonstrates superior performance to identify email servers that send spam messages [3]. It stimulates us to identify spam images with the similar supervised learning idea.

A. Data Modeling

For classification modeling, we use a corpus of 3711 unique spam images and 1999 unique ham images. Ham images seen in regular email are mostly company logos, photographs, screenshots, and cartoons. The experiments are conducted on a workstation with a Pentium M CPU at 1.73 GHz and 1 GB of memory.

Two popular classification algorithms are used in this study.

- The C4.5 algorithm for building a decision tree [4].
- The SVM algorithm for building a support vector machine [5].

C4.5 decision tree modeling is carried out in Weka, which is available at `http://www.cs.waikato.ac.nz/ml/weka/`. We use LIBSVM for SVM modeling with the RBF kernel.

TABLE VI
Confusion Matrix

	Predicted spam	Predicted ham
Real spam	TP	FN
Real ham	FP	TN

LIBSVM is available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

$$tp_{\text{rate}} = \frac{TP}{TP + FN} \tag{1}$$

$$fp_{\text{rate}} = \frac{FP}{TN + FP} \tag{2}$$

Image spam detection is by nature a cost-sensitive classification task. Table VI defines the confusion matrix. Assume that spam images are positive and ham images are negative, a False Positive (FP) is typically more expensive than a False Negative (FN). In this study, we adapt the classification modeling to this cost-sensitive scenario. The misclassification cost for a FN (denoted as fn_{cost}) is always 1 and we try different values for the misclassification cost for a FP (named fp_{cost} thereafter). We conduct a ROC analysis [6] to evaluate the effect of modifying fp_{cost} on classification modeling. Noticing that the task is to catch spam images with high confidence, we are only interested in maximizing tp_{rate} defined in (1) with $fp_{\text{rate}} \leq 1\%$ defined in (2). Hence, the ROC analysis is conducted for a maximum of 20 FPs.

Also note that both tp_{rate} and fp_{rate} are with respect to images and not email messages. Therefore, considering

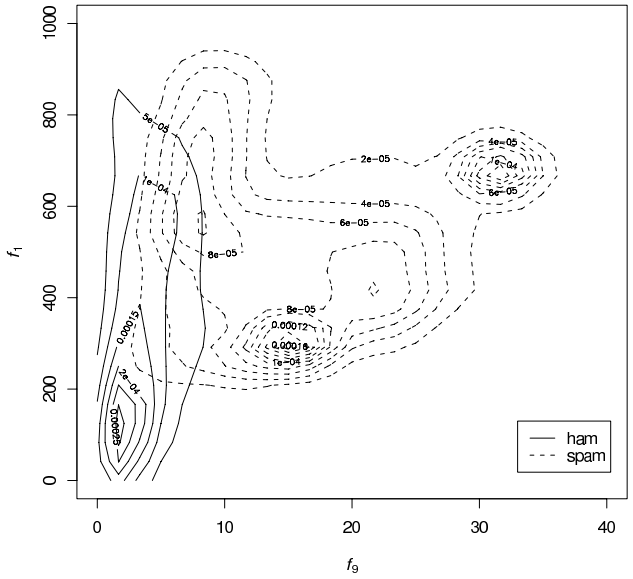

Fig. 3. Contour plot of two-dimensional probability density for f_9 and f_1.

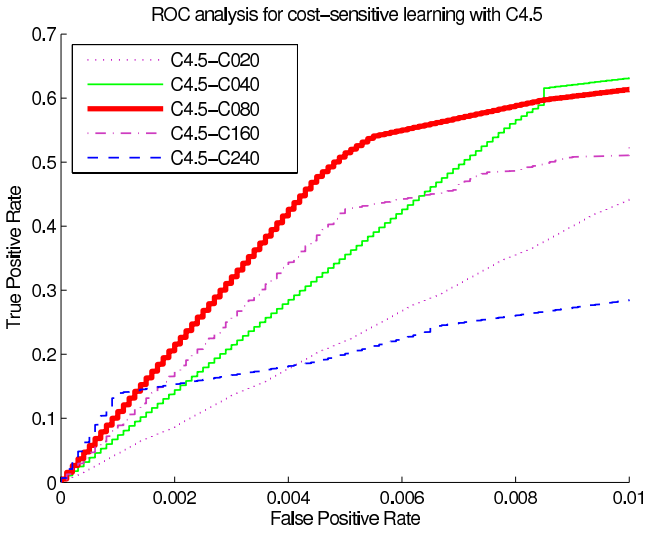

Fig. 4. ROC curves for C4.5 decision trees.

that only few ham messages contain image attachments, the per-message value for fp_{rate} will be even lower. [2]

To avoid overfitting, a stratified ten-fold cross-validation is used for classification modeling. We randomly split the corpus into 10 subsets with approximate equal size and approximate equal ratio between spam samples and ham samples. In each fold, we combine 9 of the 10 subsets to construct the training dataset. The one remaining dataset is used as the validation dataset. The training dataset is normalized so that the value of each input feature has a mean of $\mu = 0$ and a standard deviation of $\sigma = 1$. The validation dataset is normalized correspondingly. After normalization, a classifier is built on the training dataset and its predictions on the validation dataset are recorded. The validation performance can be calculated from these predictions as the estimate of the generalization performance on future unknown images.

B. Result Analysis

Figure 4 shows the results of a ROC analysis of C4.5 decision tree modeling with different fp_{cost} values. The optimal performance is achieved with $fp_{\text{cost}} = 80$ as indicated by the observation that it has the largest area under the ROC curve. It performs slightly worse than the decision tree with $fp_{\text{cost}} = 40$ for $fp_{\text{rate}} \geq 0.85\%$. Similarly, it performs slightly worse than the decision tree with $fp_{\text{cost}} = 240$ for $fp_{\text{rate}} \leq 0.13\%$. However, the tp_{rate} difference is small in these fields. We selected the decision tree with $fp_{\text{cost}} = 80$ as the best decision tree model under our constraints.

[2]Also note that most legitimate newsletters that use images use external links for these image and not image attachments.

SVM modeling yields more interesting results as shown in Figure 5. For $fp_{\text{rate}} \leq 0.20\%$ or $fp_{\text{rate}} > 0.60\%$, the best SVM is modeled with $fp_{\text{cost}} = 16$. For $0.20\% < fp_{\text{rate}} \leq 0.60\%$, the best SVM is modeled with $fp_{\text{cost}} = 64$. Based on specific requirements, we can select one of these two SVMs.

We compare the optimal classifiers in Figure 6. The three classifiers depicted are the decision tree with $fp_{\text{cost}} = 80$, the SVM with $fp_{\text{cost}} = 16$ and the SVM with $fp_{\text{cost}} = 64$. Both SVMs have a larger area under the ROC curve than the decision tree indicating superior performance. On the three different fp_{rate} levels shown in the figure, at least one of the two SVMs achieves a higher tp_{rate} than the decision tree. For example, at the 0.5% fp_{rate} level, the SVM with $fp_{\text{cost}} = 64$ can catch over 60% of the spam images.

Finally, to compare these fast classifiers to more precise but more computational expensive classifiers, we create SVM models on high-dimensional morphological feature vectors. To generate these expensive feature vectors, images are decoded and their composition is analyzed in the spatial and frequency domains. We compute SVM models for these feature vectors for both $fp_{\text{cost}} = 16$ and $fp_{\text{cost}} = 64$. Both of these SVMs are able to achieve the same performance. Figure 7 compares the performance of these expensive SVMs to the corresponding low-cost SVMs proposed in this paper in terms of tp_{rate} under the same fp_{rate} constraints. For example, these expensive SVMs can catch over 95% spams when a 0.5% fp_{rate} is acceptable.

SVM classification on low-cost feature vectors is about 200 times faster than SVM classification on morphological feature vectors. This is mostly due to the image decoding and analysis overhead introduced by the latter.

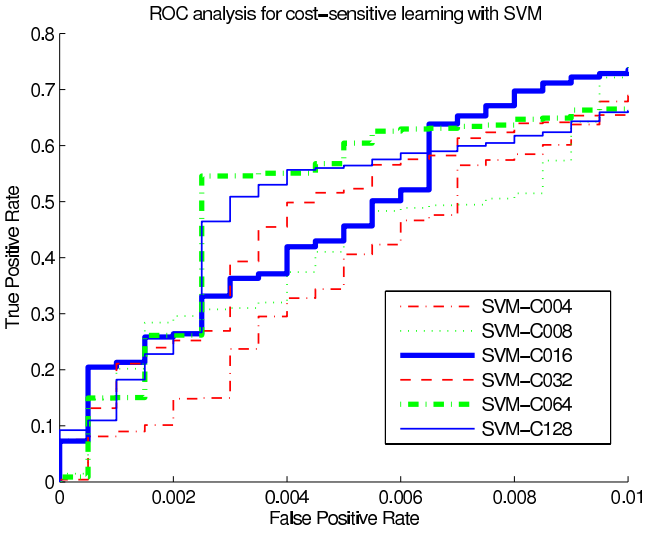

Fig. 5. ROC curves for SVM.

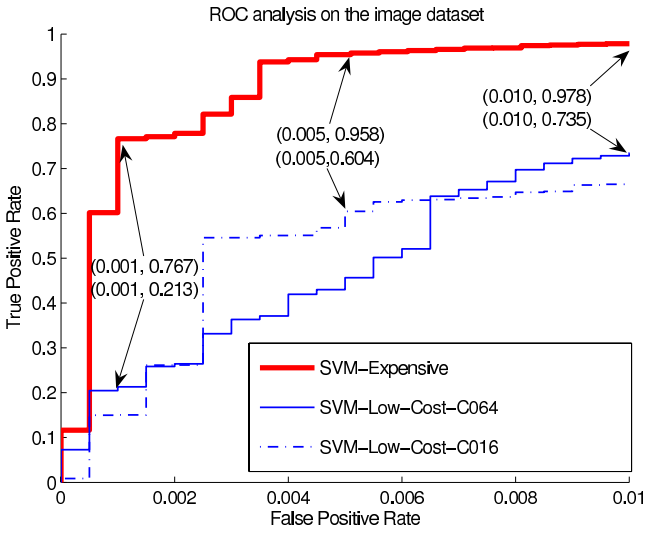

Fig. 7. ROC curve comparison for expensive and low-cost features.

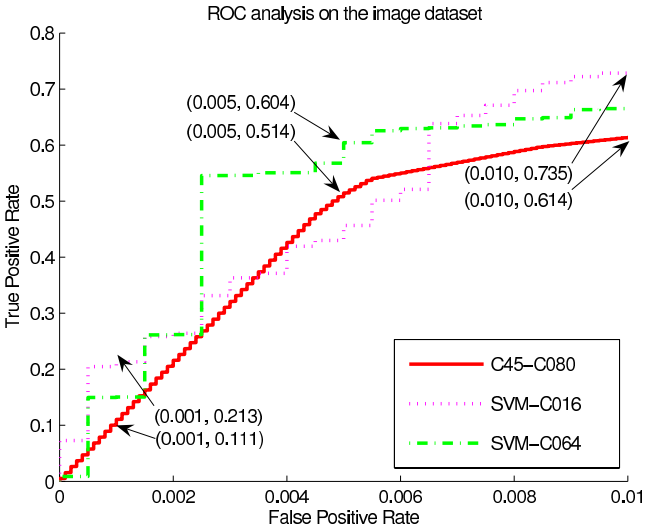

Fig. 6. ROC curve comparison.

VI. Conclusions

We propose a fast and low-cost feature extraction and classification framework to target the large amounts of image spam currently seen on the Internet. The features extraction is designed to pose a very low computational load, and the classification is biased towards a low false positive rate. About 60% of spam images can be elimated using the outlines techniques with a low false positive rate of 0.5%. Therefore, this proposed low-cost classification can effectively serve as a first tier in a multi-tier classification framework to sift out a large amount of spam images before doing expensive calculations.

Future directions include investigating the possibility to extract more features without fully decoding the image. This can be done targeted to the specific image format. For example, for JPEG images information in the EXIF record can be used. Just the presence of such a record or more detailed information like the camera type indicated in the record can be used as features. GIF images are composed of multiple blocks. The block structure extracted from a fast parse of the image can be used to generate features. Also, such a fast parse can be used to determine if the image is animated, how many frames it contains, how long each frame is displayed in the animation, and if the image is corrupted, all of which are useful features to discriminate ham from spam images.

References

[1] D. Streitfeld, "Opening Pandora's in-box," *Los Angeles Times*, May 2003. Available online at `http://www.latimes.com/technology/la-fi-spam11may11001420,1,4347344,print.story?coll=la-mininav-technology`.

[2] T. S. Furey, N. Christianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Hauessler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data.," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[3] Y. C. Tang, S. Krasser, P. Judge, and Y.-Q. Zhang, "Fast and effective spam IP detection with granular SVM for spam filtering on highly imbalanced spectral mail server behavior data," in *Proc. of The 2nd International Conference on Collaborative Computing (CollaborateCom 2006)*, 2006.

[4] R. J. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

[5] V. N. Vapnik, *Statistical Learning Theory*. New York: John Wiley and Sons, 1998.

[6] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms.," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.