IMAGE SPAM FILTERING USING FOURIER-MELLIN INVARIANT FEATURES

Haiqiang Zuo, Xi Li, Ou Wu, Weiming Hu and Guan Luo

National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China

ABSTRACT

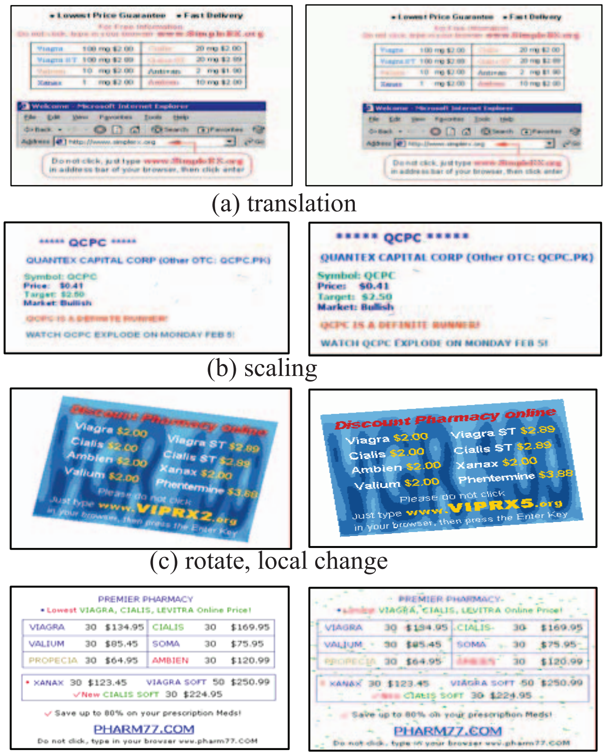
Image spam is a new obfuscating method which spammers invented to more effectively bypass conventional text based spam filters. In this paper, a framework for filtering image spams by using the Fourier-Mellin invariant features is described. Fourier-Mellin features are robust for most kinds of image spam variations. A one-class classifier, the support vector data description (SVDD), is exploited to model the boundary of image spam class in the feature space without using information of legitimate emails. Experimental results demonstrate that our framework is effective for fighting image spam.

Index Terms—Image spam, Fourier-Mellin Transform, one-class classification

1. INTRODUCTION

Email spam, also known as unsolicited bulk e-mail or junk e-mail has become a scourge for us who just want to peacefully receive and send email. Many spam-thwarting programs have been developed that inspect words, phrases, mailing histories, IP addresses, and other aspects of an email. Just as the classic battle of virus and antivirus, spammers explore new technologies in an effort to keep one step ahead of spam filters. Spammers' latest obfuscating method involves image spam, in which the main payload of the spam message is carried as an embedded image. Usually, the body of image spam contains no text or only bogus text, and the conventional text based spam filters therefore failed to detect and block it. Most of spams that break through the authors' personal anti-spam defences are image spams. Meanwhile, image spam can be more fascinating and convincing than text alone. Image spam is reported accounting for roughly 40 percent of all spam traffic now, and is still on the rise.

In recent years, many academic researchers and software security companies have turned their attention to investigating more constructive technologies to filter image spam. Fumera et al. [1] proposed an approach which exploited commercial OCR software to extract text embedded into images and then employed text categorization techniques to filter image spam. Several OCR plug-ins are also available for SpamAssassin which is a famous open-source spam filter. However, it now seems that spammers have changed their strategies. They add random dots or short lines to the background of image and apply similar CAPTCHA techniques to mislead image OCR tools. So far, very little work has been done to address this challenge. Approaches mainly differ in the set of features used to represent the image spam. Biggio et al. [2] used a low-level image feature perimetric complexity to detect whether content obscuring techniques were used, and considered that images which were obscured in a way aimed to fool OCR were likely to be spam. Dredze et al. [3] established a fast image spam detection system which used simple image features like file format, file size, image metadata, average color etc. Since most of image spams contain text, Wu et al. [4] and Aradhye et al. [5] mainly extracted embedded text features combined with other features such as color distribution or image location and trained a SVM classifier to discriminate spam images from legitimate ones.



(a) translation

(b) scaling

(c) rotate, local change

(d) adding random noises

Figure 1. Samples of different kinds of image spam.

978-1-4244-2354-5/09/$25.00 ©2009 IEEE

From the image spams we collected, we have noticed that the image spams have the following characteristics:

- Repetitiousness: Spammers tend to send the identical content many times to the same email account.
- Variability: To circumvent simple signature-based anti-spam filters, spammers usually produce many variations for a template image spam. The tricks of making image variations include: translation, rotation, scaling, local changes and adding random noises etc. Figure 1 shows the samples of different kind of image spam variations.
- Commonness: Most of image spams contain embedded text.

Based on above observation, in this paper, the Fourier-Mellin invariant descriptor which has been widely used in watermark detection and fingerprint verification systems [6] is adopted. Fourier-Mellin Transform (FMT) is a translation, scaling and rotation invariant function and performs well under noise. Figure 2 shows our framework of detecting image spam. The input image is first transformed from spatial domain to frequency domain by using a Fast Fourier Transform (FFT), and then is converted from Cartesian coordinates to Log-Polar coordinates. A second FFT, called the Mellin Transform (MT) gives the Fourier-Mellin invariant matrix, and the matrix is then stretched into a 1D vector by row concatenation. The Principal Components Analysis (PCA) is performed to project the vectors into a low-dimensional space. The final stage takes a one-class SVM algorithm to distinguish spam images from legitimate ones. Our framework makes sure that the final vectors extracted from image spam keep constant or change slightly for most image spam variations of a template.

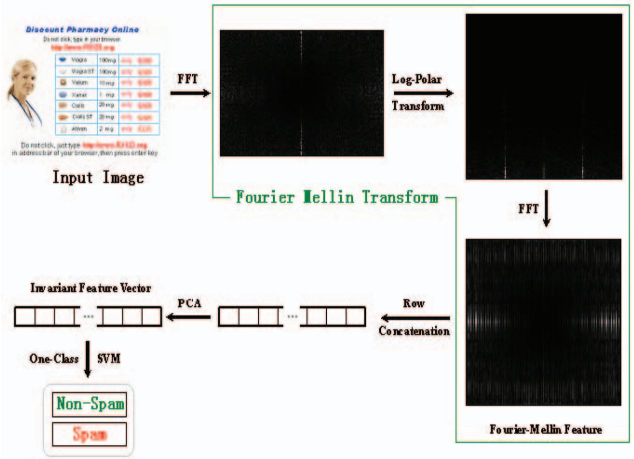


Figure 2. Overview of our framework.

The remainder of this paper is organized as follows: Section 2, 3 introduce the image spam feature extraction method and the one-class SVM classifier respectively. Section 4 demonstrates experimental results. Section 5 summarizes this paper.

2. IMAGE SPAM FEATURE EXTRACTION

In this paper, we extract Fourier-Mellin invariant features as the low-level image feature. In the following, we will demonstrate the translation, scaling and rotation invariant characteristics of Fourier-Mellin Transform.

Consider an image $g(x, y)$ that is a rotated, scaled and translated replica of $f(x, y)$,

$$g(x,y)=f[\sigma(x\cos\alpha+y\sin\alpha)-x_0,\sigma(-x\sin\alpha+y\cos\alpha)-y_0] \quad (1)$$

where α is the rotation angle, σ the uniform scale factor, and x_0 and y_0 are translational offsets. The Fourier Transform of $f(x, y)$ and $g(x, y)$ are related by

$$G(u,v)=e^{-j\varphi_s(u,v)}\sigma^{-2}[F[\sigma^{-1}(u\cos\alpha+v\sin\alpha),\sigma^{-1}(-u\sin\alpha+v\cos\alpha)]] \quad (2)$$

where $\Phi_s(u, v)$ is the spectra phase of the image $g(x, y)$. This phase depends on the translation, scaling and rotation, but the spectral magnitude

$$|G(u,v)|=\sigma^{-2}\,|F[\sigma^{-1}(u\cos\alpha+v\sin\alpha),\sigma^{-1}(-u\sin\alpha+v\cos\alpha)]| \quad (3)$$

is translation invariant.

Equation (3) shows that a rotation of the image rotates the spectral magnitude by the same angle, and that a scaling by σ scales the spectral magnitude by σ^{-2}. Rotation and scaling can be decoupled by defining the spectral magnitudes of f and g in the polar coordinates (θ, r),

$$g_p(\theta,r)=|G(r\cos\theta,r\sin\theta|,f_p(\theta,r)=|F(r\cos\theta,r\sin\theta)| \quad (4)$$

By applying some appropriate trigonometry identities, one can obtain

$$g_p(\theta,r)=\sigma^{-2}f_p(\theta-\alpha,r/\sigma) \quad (5)$$

Hence an image rotation shifts the function $f_p(\theta, r)$ along the angular axis. A scaling is reduced to a scaling of the radial coordinate and to a magnification of the intensity by a constant factor σ^2. Scaling can be further reduced to a translation by using a logarithmic scale for the radial coordinate, thus

$$g_{pl}(\theta,\lambda)=g_p(\theta,r)=\sigma^{-2}f_{pl}(\theta-\alpha,\lambda-\eta) \quad (6)$$

where, $\lambda = \log(r)$ and $\eta = \log(\sigma)$. In this polar-logarithmic representation, both rotation and scaling are reduced to translation. By Fourier transforming the polar-logarithm representations, (6),

$$G_{pl}(\zeta,\xi)=\sigma^{-2}e^{-j2\pi(\zeta\eta+\xi\alpha)}F_{pl}(\zeta,\xi) \quad (7)$$

where rotation and scaling now appear as phase shifts, and thus the normalized spectral magnitude is translation, scaling and rotation invariant. This technique decouples images rotation, scaling and translation, and can be computed efficiently by using Fast Fourier Transform [6] [7].

The resulting Fourier-Mellin feature is a $M\times N$ matrix where M and N are scale and angle resolution of the log-polar coordinate respectively and is stretched into a 1D vector of size $H= M\times N$ by row concatenation. The Principal Components Analysis (PCA) defines a transformation from R^H to a lower dimensional space R^L, $L<H$, defined by $z=W^T(x-\mu)$, where μ is the sample mean. The column vectors of W are the L eigenvectors of the covariance matrix with

largest eigenvalues. The result is a set of low dimensional vectors and is used to train our one-class SVM classifier.

3. THE ONE-CLASS SVM CLASSIFIER

One-class classification is a special type of two-class classification problem, where each of the two classes has a special meaning: the target class and the outlier class. The target class is assumed to be well sampled, and the training data reflect the area that the target data cover in the feature space, while the outlier class can be sampled very sparsely, or can be totally absent [8][10]. Spam filtering is a typical example of a problem of this type. While the spam dataset is easily accessible, a representative set of legitimate emails is difficult to collect, due to privacy concerns.

Schölkopf et al. [9] proposed an approach which is called the *ν*-support vector classifier (*ν*-SVC) and used a hyperplane to separate the target objects from the origin with maximal margin. Tax and Duin [8] proposed another approach which is called the support vector data description (SVDD) and used a hypersphere (which has a closed boundary) to contain all training objects. When all data is normalized to unit norm vectors, the SVDD is equivalent to the *ν*-SVC. In this paper the SVDD and DDtools [8][10] are used to model the image spam class.

Given a set of training target set $\{x_i\}$, i=1,…,N, the SVDD defines a model which gives a closed boundary around the data: an hypersphere. The sphere is characterized by center a and radius R and contains all training objects x_i. The structural error which has to be minimized is:

$$F(R,a) = R^2 \tag{8}$$

subject to constraints:

$$\|x_i - a\|^2 \le R^2, \quad \forall i \tag{9}$$

To allow the possibility of outliers in the training set, slack variables $\xi_i \ge 0$ are introduced and the minimization problem changes into:

$$F(R,a) = R^2 + C\sum_i \xi_i \tag{10}$$

subject to constraints that almost all objects are within the sphere:

$$\|x_i - a\|^2 \le R^2 + \xi_i, \quad \xi_i \ge 0, \quad \forall i \tag{11}$$

The parameter C gives the tradeoff between the volume of the description and the errors. By introducing Lagrange multipliers $\alpha_i \ge 0$ and $\gamma_i \ge 0$ the following Lagrangian is derived:

$$\begin{aligned} L(R,a,\alpha_i,\gamma_i,\xi_i) &= R^2 + C\sum_i \xi_i \\ &\quad - \sum_i \alpha_i\{R^2 + \xi_i - (\|x_i\|^2 - 2a\cdot x_i + \|a\|^2)\} - \sum_i \gamma_i \xi_i \end{aligned} \tag{12}$$

For each object x_i a corresponding α_i and γ_i are defined. L has to be minimized with respect to R, a and ξ_i, and maximized with respect to α_i and γ_i. By setting partial derivatives to zero gives the constraints:

$$\sum_i \alpha_i = 1 \tag{13}$$

$$a = \sum_i \alpha_i x_i \tag{14}$$

$$0 \le \alpha_i \le C, \quad \forall i \tag{15}$$

Resubstituting (13)-(15) into (12) results in:

$$L = \sum_i \alpha_i (x_i \cdot x_i) - \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \tag{16}$$

Maximizing (16) gives a set α_i. Equation (14) shows that the center of the sphere is a linear combination of the objects with weights α_i. Only objects x_i with positive weight α_i>0 are needed in the description of the data set and these objects are called the support vectors (SVs). SVs lie on the boundary (if 0<α_i<C) or outside the boundary (if α_i=C) of the sphere.

A test object z is accepted as a target object when it is inside or on the boundary of the description:

$$\|z - a\|^2 = (z \cdot z) - 2\sum_i \alpha_i (z \cdot x_i) + \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \le R^2 \tag{17}$$

R^2 is the squared distance from the center of the sphere to one of the SVs on the boundary:

$$R^2 = (x_k \cdot x_k) - 2\sum_i \alpha_i (x_i \cdot x_k) + \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \tag{18}$$

As in all formulae (Equations (16), (17) and (18)) objects x_i only appear in the form of inner products with other objects, the inner products can be replaced by a kernel function $K(x_i, x_j) = (\Phi(x_i)\cdot\Phi(x_j))$ to obtain a more flexible data description. Several kernel functions can be used, such as the polynomial kernel and the Gaussian kernel. An ideal kernel function would map the target data onto a bounded, spherically shaped area in the feature space and outlier objects outside this area. In our framework, we used an incremental version of SVDD which was adapted to cope with dynamically changing data. When new spams come, we needn't make recalculation of the entire classifier.

4. EXPERIMENTS

Two corpora of image spam were used in our experiments. One was our personal corpus and the other was the public SpamArchive corpus which used by Giorgio Fumera et al. [1] and Mark Dredze et al. [3]. Our personal corpus was made up of 1,712 images, which were collected from October 2005 to November 2007 in the authors' personal mailboxes and 11,256 valid images were extracted from the SpamArchive corpus.

To evaluate the false positive rates, the legitimate image email (also called non-spam or ham) corpus is needed. Note that our one-class classifier is able to work, solely on the basis of spam corpus. However, the legitimate image email corpus is difficult to collect due to the following reasons:

- Legitimate emails containing images are much rarer than spam ones.
- Legitimate emails usually contain personal information, and it is not easy to distribute them.

Instead, we chose to use two subsets of Caltech-256 (www.vision.caltech.edu) as non-spam corpora for our experiments. The Caltech-256 contains 30,608 images of 257 Categories. We randomly selected 10 images per category, and our first subset contained 2570 images. The second subset contained 15,304 images, and about half images were selected each category. The fist subset was used to evaluate the false positive rates of the classifier trained by our personal image spam corpus, and the second subset was used to evaluate the false positive rates of the classifier trained by the SpamArchive corpus. The datasets are summarized in Table1.

Table 1. A summary of our datasets.

Experiments	Corpus	Number of Images
Experiment I	Personal Spam	1712
	Caltech-256 Subset I	2570
Experiment II	SpamArchive Spam	11256
	Caltech-256 Subset II	15304

We assessed our method by using 10-fold cross-validation. The spam corpus was randomly divided into 10 folds, and one fold was left together with the non-spam corpus as the test set and the other folds were used for training. The experiment was repeated 10 times, and the classification result is calculated by averaging over 10 runs.

Table 2. Experimental results.

Experiments	Precision	Recall	*F*1
Experiment I	98.92%	83.65%	90.64%
Experiment II	98.74%	78.60%	87.53%

By adjusting the kernel parameters and the predefined target rejection rate on the training set, we have gained a high precision and an acceptable recall. This is because that most users would rather receive more spams than lose a useful legitimate email.

Our algorithm is very fast, the Fourier-Mellin invariant features can be computed efficiently by using twice Fast Fourier Transform (FFT). The average time to classify an image is 190 milliseconds on our personal computer.

It is not easy to compare other previously proposed techniques with our proposed algorithm directly due to different datasets and measures that were used. Table 3 shows the performance of [5]. The detection rate is equivalent to the recall measure that we used. The spam dataset SPAM-1 and SPAM-2 which they used contained 497 and 1245 images respectively, and the legitimate images were collected from Google Image Search.

Table 3. Experimental results [5].

Dataset	Detection Rate	False Positive
SPAM-1	76.25%	6.5%
SPAM-2	82.75%	17.25%

5. CONCLUSIONS

A framework for filtering image spams by using the Fourier-Mellin invariant features is presented in this paper. Fourier-Mellin features are robust for most kinds of image spam variations. The SVDD classifier can distinguish spam images from legitimate ones efficiently solely on the basis of spam corpus. Experimental results demonstrate that our framework is effective for fighting image spam.

6. ACKNOWLEDGMENT

As the www.spamarchive.org has been offline, the authors thank Giorgio Fumera and Mark Dredze for making the SpamArchive corpus available. This work is partly supported by NSFC (Grant No. 60825204 and 60672040) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453), and Youth Science Foundation of CASIA.

7. REFERENCES

[1] G. Fumera, I. Pillai and F. Roli, “Spam filtering based on the analysis of text information embedded into images,” *Journal of Machine Learning Research*, pp. 2699-2720, 2006.

[2] B. Biggio, G. Fumera, I. Pillai, and F. Roli, “Image Spam Filtering by Content Obscuring Detection,” *Fourth Conference on Email and Anti-Spam*, Mountain View, CA, USA, 2007.

[3] M. Dredze, R. Gevaryahu, A. Elias-Bachrach, “Learning Fast Classifiers for Image Spam,” *Fourth Conference on Email and Anti-Spam*, Mountain View, CA, USA, 2007.

[4] C.T. Wu, K.T. Cheng, Q. Zhu and Y.L. Wu, “Using visual features for anti-spam filtering,” *In Proceedings of the IEEE International Conference on Image Processing*, pp. 501-504, 2005.

[5] H.B. Aradhye, G.K. Myers and J.A. Herson, “Image Analysis for Efficient Categorization of Image-based Spam E-mail,” *International Conference on Document Analysis and Recognition*, pp. 914-918, 2005.

[6] T.B.J. Andrew, N.C.L. David, T.S. Ong, “An efficient fingerprint verification system using integrated wavelet and Fourier-Mellin invariant transform,” *Image and Vision Computing*, pp. 503-513, 2004.

[7] Q.S. Chen, M. Defrise, and F. Deconinck, “Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition,” *Pattern Analysis and Machine Intelligence*, pp.1156-1168, 1994.

[8] D.M.J. Tax and R. Duin, “Support Vector Data Description,” *Machine Learning*, pp.45-66, 2004.

[9] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, “Estimating the Support of a High Dimensional Distribution,” *Neural Computation*, pp. 1443-1471, 2001.

[10] D.M.J. Tax, *DDtools, the Data Description Toolbox for Matlab*, version 1.6.1, 2007.