Spam Image Discrimination using Support Vector Machine based on Higher-Order Local Autocorrelation Feature Extraction

Hongrong Cheng, Zhiguang Qin, Qiao Liu, Mingcheng Wan
School of Computer Science & Engineering
University of Electronic Science & Technology of China,
Chengdu, Sichuan 610054 China

Abstract— **In this paper, a new method is proposed for discriminating spam images from non-spam images. This method extracts edge features of a binarized image by using higher-order local autocorrelation(HLAC), and then input those features to support vector machine (SVM) for classification. Our method has three unique characteristics. First, the method extracts edge features which can represent major edge properties of an image without limitations imposed by image edges' directions or distributions. Second, the method can tolerate effectively slight changes of color, texture, size, layout of an image, and characteristics of text embedded in it. Third, the method is fast because of no time cost of text location and recognition. Experimental results for the public personal dataset show that the proposed method can separate spam images from non-spam images with minimum recognition rates of 98%.**

I. INTRODUCTION

As a cheap and easy means for exchanging messages, email has gained enormous popularity. At the same time, the volume of spam (also known as unsolicited email) sent on daily basis poses a great threat to the regular utility of email communications [1]. To filtering spam, text categorization techniques have been investigated in machine learning community in the past ten years. Text-based learning filters have grown in sophistication and effectiveness in detecting email spam [2], [3], [4]. To defeat such techniques, spammers recently began to embed the spam textual content into attached images which are typically called as spam images (see the examples in Fig. 1). While the textual contents in such images can be normally read by receivers, the image spam is shielded from text-based anti-spam filters. The rapid spread of image spam presents a great challenge to most of existing spam filters in terms of discrimination performance [5].

Recently, several attempts have been made to address identifying spam images by utilizing specific features of images, such as [6], [7], [8]. However, all of the existing proposals suffer from one obvious drawback. Slight changes in an image, such as the introduction of pixel noise or randomization of the color palette, texture, sizes or characteristics of text embedded in the images, can have great ramifications for the extracted features, making those proposals infeasible to ensure detection performance in a real environment. For example, [8] detects spam image by using optical character recognition (OCR) system to convert spam images back to text for processing by text-based filters. The OCR-based approaches can be effective against spam image for cases in which no content obscuring techniques are used by spammers. However, spammers have started to obscure image text to defeat OCR tools by exploiting to CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Human Apart)-like techniques (see the example in Fig. 1 most left bottom). In [7], [9], the embedded-text features, banner and graphic features, and image location features are extracted as spam images' indicative properties. However, the accuracy of text region extraction is still sensitive to obscuring techniques. Moreover, according to our analysis of thousands of attached images in emails, we find that many non-spam images have similar those features with spam images. Since the proposed properties are not distinctive enough, they result in unsatisfactory discriminant performance. In [10], a method based on edge direction (ED) and edge orientation autocorrelogram (EOAC) is presented. It builds the ED histogram and EOAC matrix to describe the shape of images. To our best knowledge, it is one of few existing spam image identification approaches which are related to higher semantic content of images without time cost of text location and recognition. However, this method makes sense only when the area of embedded-text region covers dominating proportion of an image (not less than 50%).

The primary objective of this paper is to present a new approach for discriminating spam images from non-spam images. The method computes 25 normalized local edge features based on HLAC and then serves those features as input to the SVM classifier which is applied to map the input space into a possibly high-dimensional feature space and then generalize the optimal hyperplane with maximum margin between spam image class and non-spam image class. The motivation of using HLAC feature extraction is based on the following considerations: (1) the content of an image is inherently related to its local edge autocorrelation features, (2) autocorrelation function is shift-invariant [11] and can be calculated quickly [12]. Our proposed method not only can represent major edge properties of an image without limitations respect to edge directions or distributions, but also can be insensitive to slight changes of color, texture, size, layout of an image, and characteristics of text embedded in it. In addition, the method is fast without

978–1–4244–1674–5/08/$25.00 © 2008 IEEE

CIS 2008

spending any time on text location and recognition, which makes it possible for our approach to operate on heavily loaded email servers. We evaluate the performance of the proposed method on a public collection of images described in [13]. The results show that the proposed approach separates spam images from non-spam images with recognition rates of above 98%.

This paper is organized as follows. Section 2 gives a brief overview of some related work. Section 3 details the feature extraction based on HLAC. In Section 4, we describe the spam image discrimination using the HLAC-SVM aggregation. Experimental results are carried out in Section 5. Finally, we summarize our contributions in Section 6.

II. Related Work

Several approaches have been developed to address identifying spam images by utilizing specific features of images in recent years [6], [7], [10], [13], [16]. Since overlaid text contains important information about image content, some attempts have been made by using OCR-based techniques [8]. While these solutions promises to detect image spam with a certain level of accuracy, the existing OCR-based algorithms suffer from three following difficulties: (1) the location and recognition of text is never perfect; (2) they are unable to identify images obscured by CAPTHA-like techniques; and (3) they are computationally expensive and thus cannot operate on heavily loaded email servers. Instead of recognizing full text, the text-region-based methods [6], [7], [9] only extract approximate regions with overlaid text from images and generate several simple features based on the extracted text regions. The classifiers in [13], [14], [15] extract features by simple analysis of image elements (e.g., color saturation, color heterogeneity, texture, etc.). However, since those features are related on relatively generic characteristics of images, they are not distinctive enough to achieve satisfactory discriminant capacity. Becides, the text-region-based approaches requires text location, which is also time consuming. The approach described in [16], [17] recognizes image spam based on detecting the presence of obscuring techniques. In a work by N.P. Nhung [10], the authors relied on ED and EOAC of an image to present its shape. This method requires ten sets of templates constructed with text regions covering from 50% to 90% of the whole images.

III. Feature Extraction based on Local Edge Autocorrelation

Embedded text can be located anywhere within an image. To make the features insensitive to location change of embedded text in an image, we need extract shift invariant features from images. It is well known that the autocorrelation function is shift-invariant [11], [18]. The Nth-order autocorrelation functions, extensions of autocorrelation functions, are defined as

$$x_f^N(a_1,...a_N)=\int_p f(r)f(r+a_1)...f(r+a_N)dr,$$

(a) Simple Background

(b) Complex Background

Fig. 1. Examples of spam images from the public personal dataset

(a) Photos (b) Logos (c) Cartoons

(d) Maps (e) Directions (f) Handwriting

Fig. 2. Examples of non-spam images from the public personal dataset

where $f(r)$ denotes the intensity at the reference point r, and $a_1, ..., a_N$ are N displacements.

In our approach, feature extraction is based on binarized edge images. Let a local mask binarization edge plane be demoted by P. $f(r)$ corresponds to binarization edge characters of an image on P, with value 1 or 0. Since P is a discrete value, the functions are defined as

$$x_f^N(a_1,...a_N)=\sum_P f(r)f(r+a_1)...f(r+a_N),$$

where $r \in P$ and the support of f is include in P, defined as $Supp(f) = \{r|f(r) > 0\}$, $Supp(f) \subset P$. Then a shift of $f(r)$ within P is represented by $S(a)f(r) = f(r+a)$, where the displacement $a \in R^2$ is restricted so that the support does not exceed P. Let $x[f]$ denote a feature of the binarization edge of an image $f(r)$ extracted over P. One requirement of $x[f]$ is the shift invariance of $x[f]$, represented by $x[S(a)f] = x[f]$, for $\forall x \in R$, $Supp(S(a)f \subset P)$. Another requirement is the additivity of $x[f]$, represented by $x[f_1 + f_2] = x[f_1] + x[f_2]$ for $Supp(f_1) \cap Suppf_2 = \phi$. The orders and displacements are arbitrary. However, higher-order features with a large displacement region become extremely numerous. Hence we restrict the order N up to the second (that is $N = 0, 1, 2$) for practical application. Considering

the correlations of closed values are much higher than the correlations between far points, we also restrict the size of P to 3×3. By eliminating the displacements which are equivalent by the shift, the number of the masks of the displacements is reduced to 25 as shown in Fig.3. The center of each mask is the reference point, each filled cell stands for "1", which means there is an edge pixel, and each blank cell stands for "don't care".

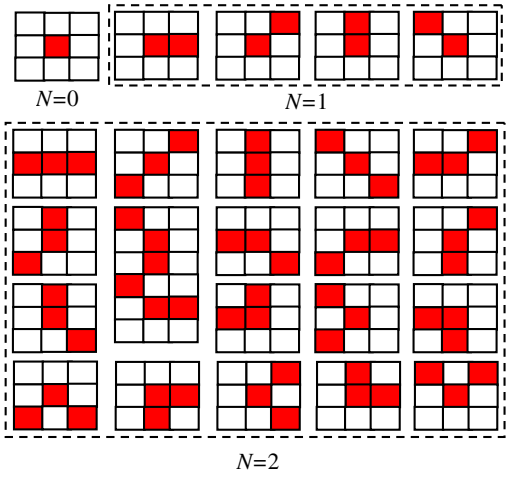

Fig. 3. Local mask patterns for computing HLAC features

The features are calculated by scanning a binarized edge image with the 25 local 3×3 mask binarization edge patterns and by computing the sums of the products of the values of the corresponding pixels to "1" in the mask patterns. If a displacement exceeds the image data while scanning the boundary of an images, the row and the column, which are filled in 0, are added respectively. Let the features of an image i be denoted as $x_{i1}, ..., x_{i25}$, the feature vector is defined by $x_i = (x_{i1}, ..., x_{i25})$. Let the reference point be (j, k), then $x_{i1}, ..., x_{i25}$, are defined as $x_{i1}=\sum_j \sum_k f(j,k)$, ..., $x_{i25}=\sum_j \sum_k f(j,k) f(j-1, k-1) f(j+1, k-1)$.

Based on HLAC, we obtain such features that represent major edge properties of an image. The 25 features are shift invariant. This is a desired characteristic in our case because text can be located anywhere within an image. Since the total number of edge pixels depends on the total number of image pixels, image scaling affects the total number of edges. To make the features against scaling variation, the 25 feature histogram is normalized with respect to the sum of the number of edge points of each mask for an image. Hence, the normalized HLAC features become invariant to linear scaling of an object in the original image. As a result, the binarized edge features are insensitive to slight changes of color, texture, size, layout of an image, and characteristics of text embedded in it. By combing these features with classifier which decide whether an image is spam or not, we can design an effective spam image filter.

IV. Spam Image Discrimination Using HLAC-SVM

The support vector machine (SVM) is a classification and regression algorithm which was developed by Vapnik [19] and it is gaining popularity due to many attractive features and promising empirical performance.

In this work, we use the normalized edge features with 25 dimensions as input to the SVM for classification. For a two-class pattern recognition problem, the main idea of SVM is to construct a nonlinear kernel function to map the data from the input space into a possibly high-dimensional feature space and then generalize the optimal hyperplane with maximum margin between two classes. Having found such a hyperplane, the SVM predicts the label of a new example by mapping it into the feature space and deciding on which side of the hyperplane the example is located. Below we describe the task of SVM classifier briefly:

Let there be a labeled training set $\{x_i, y_i\}_{i=1}^n \subset R^d \times \{-1, +1\}$, where d is the dimension of sample x_i, n is the number of the samples in the training set. Each y_i is either 1 or -1 depending on the class of sample x_i. In general, 1 and -1 correspond to positive and negative training samples respectively. In the linear separable case, the decision function can be written as $f(x)=\sum_i a_i y_i x \cdot x_i + b = x \cdot \sum_i a_i y_i x_i + b = x \cdot w + b$, where $a_i \geq 0$, b is a bias term, w is the normal vector of the classification hyperplane and . is a dot-product operator. Typically, the multipliers a_i have non-zero values only for a small subset of the training set, which is called the support set and its elements the support vectors. The optimal hyperplane is found so as to maximize the sum distance to the closest positive and negative training samples. The distance is called margin and the optimal hyperplane is obtained by maximizing $\frac{2}{\|w\|^2}$ subject to a set of constraints. By relaxing the constraints, the non-separable case can also be handled and the optimization problem can be formulated as:

$$\begin{aligned} &min.\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ &s.t.\ y(w \cdot x + b) \geq 1 - \xi_i, \end{aligned} \tag{1}$$

where $C \geq 0$ is a user-defined penalty paprameter of the error term, and the slack variables ξ_i take non-zero values for points that are misclassified.

In our work, given a set of labeled images $(x_i, y_i), ..., (x_n, y_n)$, x_i is the feature representation of one image, and y_i is the class label (-1 denotes negative and +1 denote positive). Training the HLAC-SVM classifier leads to the mapping $R^{25} \rightarrow \{-1, +1\}$, where R^{25} represents the normalized 25 HLAC features of sample x_i, n is the number of the samples in the training set. Each y_i is either 1 or -1 depending on the class of sample x_i. The HLAC-SVM classifier is trained by solving the quadratic optimization problem formulated in (1). The mapping from the input space to the feature space is done by using kernel functions. Let $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$ be such a kernel function, where the training samples $X = \{x_1, ..., x_n\}$ are mapped into a higher dimensional space by the function Φ. The four types

of kernel functions frequently used with SVM are formulated as follows:

$$\begin{aligned} \text{Linear} : K(x_i, x_j) &= x_i^T x_j \\ \text{Polynomial} : K(x_i, x_j) &= (\gamma x_i^T x_j + r)^d, \gamma > 0 \\ \text{RBF} : K(x_i, x_j) &= exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \\ \text{ANOVA} : K(x_i, x_j) &= (\sum_{k=1}^{n} exp(\gamma(x_i - x_j)^2)^d \end{aligned} \quad (2)$$

Here, γ, r, d are kernel papameters. So far, it is difficult to decide on which kernel is best to select for a particular problem. To choose the most appropriate kernel function among the four kernels formulated in (2), we carry out classification experiments in each kernel function and make comparisons of the results.

V. Experiments

In this section, we present some experiments for the public personal image dataset to demonstrate the effectiveness of the proposed method. Three typical metrics are used for the performance evaluation. To conduct experiments, we use winSVM [20], a windows implementation of support vector machine. It provides the optimization feature to help facilitate parameter selection and analysis. The dataset, evaluation metrics, and experimental results are detailed as follows:

A. Dataset

Unlike the situation with text-based spam filtering, there is few public spam image/non-spam image datasets for benchmark nowadays. Our experiments are carried out by using the collection presented in [13], available at http://www.seas.upenn.edu/~mdredze/datasets/image_spam/. The original dataset includes 3298 color spam images and 2021 color non-spam images with different size. To make the experiments more reasonable, we deal with the original dataset by two steps. First, we remove images suffering from the following cases: (1) images smaller than 10×10 pixels since our observation shows that those images are often used as blank spacers in HTML documents rather than real images delivering spam information; (2) spam images without any embedded text (for example, the photographs of people, the picture of a pen) because it is difficult even for human beings to judge those images as spam or non-spam without the help of semantic representation delivered by embedded text, let alone for machines. Second, we ignore some images which cannot be processed successfully by our image reader. Finally, we focus on a dataset which contains 3114 spam images with sizes from 105×295 to 1180×300 pixels and 1699 non-spam images with sizes from 12×12 to 8727×1434 pixels. Most are GIF images, and the rest are BMP, JPEG or PNG images. About half images in the dataset contain simple background (such as Fig. 1a, Fig. 2f); others contain graphic and photographic elements with different levels of complexity (such as Fig. 1b, Fig. 2e).

TABLE I
Recognition Rates Related Metrics

	Real spam(t=1)	Real non-spam(t=-1)
Predicted spam(y=1)	true positive(TP)	false positive(FP)
Predicted non-spam(y=-1)	false negative(FN)	true negative(TN)

B. Performance Evaluation Metrics

In our experiments, we assume that spam images are positive and non-spam images are negative. For evaluation of the proposed approach, the performance evaluation metrics defined in Table I and the following formulations are used:

$$\begin{aligned} Accuracy = p(t = 1, y = 1) &= \frac{TP + TN}{TP + TN + FP + FN} \\ Precision = p(t = 1 | y = 1) &= \frac{TP}{TP + FP} \\ Recall = p(y = 1 | t = 1) &= \frac{TP}{TP + FN} \end{aligned} \quad (3)$$

C. Results

We assess the accuracy, precision and recall of the SVM classifier in the four kernels by using 10-fold cross validation. The dataset is randomly divided into 10 folds of equal size. For each run, one fold is left as the test set and the other folds are used for training. For each metric of the classifier in a specific kernel, the experiment is repeated 10 times with different folds being the test sets.

For classification accuracy, the comparison in Fig. 4 shows that the ANOVA and RBF kernels have clearly higher accuracy than the linear and polynomial kernel. The polynomial kernel performances worst, with unstable accuracy for different test sets, from 71% to 97%. For classification precision and recall, the comparisons in Fig. 5 and Fig.6 show the similar results with that in Fig. 4, i.e., the ANOVA and RBF kernels outperform the linear and polynomial kenrels. The polynomial kernel shows unstable performace not only for a specific metric in one 10-fold cross validation (such as for accuracy in Fig. 4), but also for different metrics in a specific test set. For example, in the 9th test(indicated by the points corresponding to "9" value in x-axis in the figures), while the polynomial kernel shows relatively high precision (as shown in Fig. 5), it has poor accuracy (as shown in Fig. 4) and recall (as shown in Fig. 6). The linear kenel presents the medium level of performance.

The mean values of accuracy, precision and recall are calculated by averaging over 10 runs for each kernel. The results in Table II show that as expected, the best results are obtained by the classifier in ANOVA or RBF kernel, which separates spam images from non-spam images with accuracy of not less than 98%.

VI. Conclusion

This paper presents a new approach for separating spam images from non-spam images. This method can represent the major edge properties of an image without limitations respect to edge directions and distributions. Besides, it is insensitive

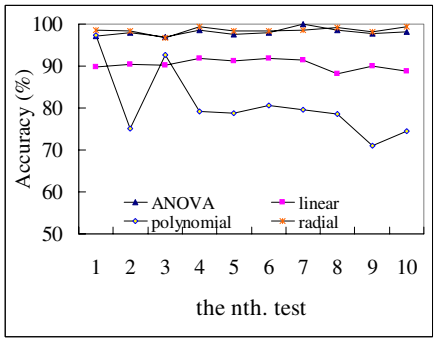

Fig. 4. Classification accuracy for the four kernels

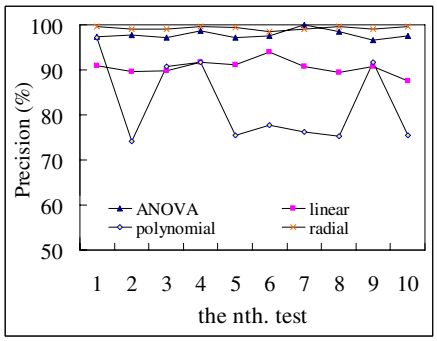

Fig. 5. Classification precision for the four kernels

to slight changes of color, texture, size, layout of an image, and characteristics of text embedded in it. Furthermore, the method is fast without spending any time on text location and recognition. The experimental results for the public personal image dataset show that the SVM classifier in ANOVA or RBF kernel can discriminate spam images from non-spam images with recognition rates of above 98%.

ACKNOWLEDGMENT

The authors are indebted to anonymous reviewers for useful comments. The first author also would like to thank her colleague Hua Yan for the useful discussions.

This research is supported by National High Technology Research and Development Program of China (863 Program) under Grant 2006AA01Z411.

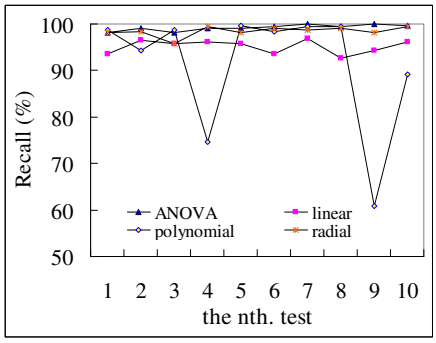

Fig. 6. Classification recall for the four kernels

TABLE II
RESULTS USING DIFFERENT KERNELS(%)

	ANOVA	Linear	Polynomial	Radial
Accuracy	98%	90%	81%	98%
Precision	98%	90%	83%	99%
Recall	99%	95%	91%	98%

REFERENCES

[1] Symantec Inc. *The state of Spam A monthly report-January 2008*. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/Symantec_Spam_Report_-January_2008.pdf.

[2] M. Sahami, S. Dumais, D. Heckerman and E. Horvitz, *A Bayesian approach to filtering junk e-mail*, in Proc. AAAI Workshop on Learning for Text Categorization pp. 55-62, 1998.

[3] H. Drucker, D.Wu, and V.N. Vaphik, *Support Vector Machines for Spam Categorization*. IEEE Transactions on Neural Networks, Vol.20(5), pp. 1048-1054, 1999.

[4] X. Carreras and L. Marquez, *Boosting Trees for Anti-Spam Email Filtering*, in Proc. of 4th International Conf. on Recent Advances in Natural Language Processing, . RANLP-2001, 2001.

[5] C. Pu and S. Webb, *Observed trends in spam construction techniques: A case study of spam evolution*, in Proc. of the 3rd Conf. on E-Mail and Anti-Spam, 2006.

[6] H.B. Aradhye, G.K. Myers, and J.A. Herson, *Image Analysis for Efficient Categorization of Image-based Spam E-mail*, in Proc. of ICDAR. Seoul, Korea,pp. 914-918, 2005.

[7] C.T. Wu, *Embedded-text detection and its application to anti-spam filtering*. Master's thesis, University of California- Santa Barbara, 2005.

[8] G. Fumera, I. Pillai, and F. Roli, *Spam filtering based on the analysis of text information embedded into images*, Journal of Machine Learning Research (special issue on Machine Learning in Computer Security), 7:2699-2720, 2006.

[9] C.T. Wu, K.T. Cheng, Q. Zhu, and Y.L. Wu, *Using visual features for anti-spam filtering*, IEEE International Conf. on Image Processing, 2005.

[10] N.P. Nhung and T.M. Phuong, *An Efficient Method for Filtering Image-based Spam E-mail*, Heidelberg, Berlin: Springer-Verlag, LNCS 4673, pp. 945-953, 2007.

[11] V. Popovici and J.P. Tguran, *Higher Order Autocorrelations for Pattern Classification*, pp.724-727, 2001.

[12] K. Yamamoto, I. Ishii, *A design of higher order auto-correlation vision chip*, IEICE Trans. Inf. Sys. (D-II) J86-D-II(8) pp.1205-1211 (in Japanese), 2003.

[13] M. Dredze, R. Gevaryahu, and A.E. Bachrach, *Learning Fast Classifiers for Image Spam*,in Proc. of the Conf. on Email and Anti-Spam (CEAS), 2007.

[14] S. Krasser, Y. Tang, J. Gould, D. Alperovitch, and P. Jude, *Identifying Image Spam based on Header and File Properties using C4.5 Decision Trees and Support Vector Machine Learning*, in Proc. of the 2007 IEEE Workshop on Information Assurance, pp. 255-261, 2007.

[15] B. Byun, C.H. Lee, S. Webb, and C.Pu, *A Discriminative Classifier Learning Approach to Image Modeling and Spam Image Identification*, in Proc. of the 4th Conf. on Email and Anti-Spam, Mountain View, California USA, 2007.

[16] B. Biggio, G. Fumera, I. Pillai, and F.Roli, *Image Spam Filtering by Content Obscuring Detection*, in Proc. of the 4th Conf. on Email and Anti-Spam, Mountain View, California USA, 2007.

[17] B. Biggio, G. Fumera, I. Pillai, and F. Roli, *Image Spam Filtering Using Visual Information*, in Proc. of 14th international Conf. on Image Analysis and Processing, 2007.

[18] J.A. McLaughlin and J. Raviv, *Nth-order autocorrelations in pattern recognition*, Information and control, vol.12, pp.121-142, 1968.

[19] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verleg, 1998.

[20] S. Rping, *mySVM-Manual*, University of Dortmund, Lehrstuhl Informatik 8, http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/ M. Sewell, 2000.