Image Spam Filtering by Content Obscuring Detection

Battista Biggio, Giorgio Fumera, Ignazio Pillai, Fabio Roli
Dept. of Electrical and Electronic Eng., University of Cagliari
Piazza d'Armi, 09123 Cagliari, Italy
{bat,fumera,pillai,roli}@diee.unica.it

ABSTRACT

We address the problem of filtering *image spam*, a rapidly spreading kind of spam in which the text message is embedded into attached images to defeat spam filtering techniques based on the analysis of e-mail's body text. We propose an approach based on low-level image processing techniques to detect one of the main characterstics of most image spam, namely the use of content obscuring techniques to defeat OCR tools. A preliminary experimental evaluation of our approach is reported on a personal data set of spam images publicly available.

1. INTRODUCTION

Current spam filters are made up by several modules devoted to analyze different components of e-mails (sender's address, header, body, attachments), to detect specific characteristics of spam e-mails. The outputs of the different modules are then combined to get the final decision about the "spamminess" of an e-mail. A typical example of this architecture is the open-source filter SpamAssassin.[1] In the past ten years the machine learning research community has been interested in the spam filtering task, and in particular in the analysis of e-mail's textual content, formerly carried out by simple hand-made filtering rules like keyword detection, which were easily circumvented by spammers. Text categorization techniques with potentially higher generalization capability have been developed (see for instance [10, 5, 7, 1]) and are now used in many spam filters. To defeat such techniques, recently spammers started to embed the spam message into attached images (often bogus text is also inserted into e-mail's body), a trick known as *image spam* (see the examples in Figure 2). The rapid spread of image spam leads us to argue that in the near future computer vision and pattern recognition techniques will gain a prominent role against image spam.

Using OCR tools to extract text embedded into images, and processing it using text categorization techniques was thoroughly investigated by the authors in [6]: it was found that this approach can be effective for clean images. A much simpler approach based on keyword search on text extracted by OCR was recently implemented in a plug-in of the SpamAssassin filter.[2] However very recently spammers started to "obscure" image text to defeat OCR tools (as can be seen from the examples in Figure 2). To this aim, it is worth noting that spammers could exploit to their advantage techniques used to create CAPTCHAs (see Figure 2, bottom), which were introduced just to defend against robot spamming. Although we believe that content obscuring techniques are not likely to be used in every kind of spam (for instance, *phishing* e-mails should look as if they come from reputable senders, and thus should be as "clean" as possible), so that spam filtering modules based on OCR techniques can still be useful, we also argue that different techniques should be devised for the cases in which noise introduced by spammers il likely to make standard OCR tools ineffective.

To our knowledge, few works in the scientific literature dealt with image spam so far [2, 11], and no one addressed the specific issue of noisy text images, although some vendors have already included in their filters image processing modules. In this work we propose a specific approach aimed at detecting the presence of noisy text due to the use of content obscuring techniques against OCR tools. This can be viewed as complementary to the approach based on OCR tools investigated in [6], since it aims at detecting the "noise" (the adversarial clutter contained in the image) instead of the "signal" (the text embedded into the image). The presence of noisy text can be used by a spam filter as an indication of e-mail's spamminess, to be properly combined with the outputs provided by other filtering modules to reach a final decision. An overview of our approach is given in Section 2. In Section 3 we describe a possible implementation we are currently investigating, based on detecting the similar effects of many content obscuring techniques on binarized images. In Section 4 a preliminary experimental analysis of our approach is reported, on a personal data set of spam images publicly available.

2. PROPOSED APPROACH

In this paper we propose an approach against the kind of image spam in which obscuring techniques are used by spammers to make the embedded text unreadable by OCR tools. Our approach consists in developing techniques to detect the presence of noisy text into an image. Such techniques are intended to be used by a specific module of a spam filter, whose output could be either a crisp label indicating the presence or absence of noisy text, or a real num-

CEAS 2007 - Fourth Conference on Email and Anti-Spam, August 2-3, 2007, Mountain View, California USA

[1] http://spamassassin.apache.org

[2] http://wiki.apache.org/spamassassin/BayesInSpamAssassin

ber indicating the "amount" of noise in a proper scale (for instance in the range $[0, 1]$, where 0 means that the image is clean). The rationale is that the presence of noisy text can be considered as an indication that an e-mail is spam. However this can not give the certainty that the e-mail is spam, since noisy text could also be present in legitimate images. This is the reason why we do not intend the output of such a module to be directly a spam/legitimate label, or the likelyhood (or probability estimate) that the e-mail is spam. Instead, we intend such output as providing a piece of information which will subsequently have to be combined with the outputs provided by other modules to reach a final and hopefully more reliable decision, as happens in current filters. As a trivial example, if an e-mail has an attached image with noisy text but the sender address is in the white list, it can be labelled as legitimate by the spam filter.

The techniques proposed in this work to detect the presence of noisy text into an image are based on the following idea. Different kinds of content obscuring techniques can be used against OCR (adding background noise interfering with text, distorting text lines or single characters, etc.), including methods developed for building CAPTCHAs, and many of them have already been observed (see Figure 2). Clearly, methods tailored to detect specific obscuring techniques are likely to exhibit a poor generalization capability and to be easily circumvented by spammers. However we observed that several obscuring techniques like the ones in Figure 2 (except for the one at the bottom-right) turn out to compromise OCR effectiveness by producing similar effects on the binarized image (note that OCR tools work by first binarizing images to divide characters from background). In particular, these effects consists in broken characters, characters interfering with background noise components, and large background components overlapping characters. Accordingly, we argue that many obscuring techniques (although not all of them, for instance the ones based on character distortion as at the bottom-right of Figure 2) can be detected by looking for the kind of image defects mentioned above in the binarized image. In the rest of the paper we will discuss a method we are developing to detect such kind of noise and to measure its amount.

3. NOISY TEXT DETECTION

The problem we are addressing exhibit some analogies with the problem of measuring image *quality* addressed in the OCR literature (see for instance [4]). However these methods are based on specific assumptions which do not hold in our task (like equally sized characters). An interesting suggestion came instead to us from the "BaffleText" CAPTCHA [3], which uses random masking to degrade text images resulting in image defects similar to the ones we are interested in. In [3] the *complexity* of an image for a human reader (not for an OCR) was evaluated through the *perimetric complexity* measure used in the psychophysics of reading literature [9], defined as the squared length of the boundary between black and white pixels (the "perimeter") in the whole image, divided by the black area, P^2/A. In [3] P^2/A was computed on the whole image, but this does not fit our goals since, for instance, the P^2/A value depends on the number of characters into an image. However, taking into account that P^2/A is scale-invariant, we found that measuring the P^2/A value of each *individual* component of a binarized image can allow to characterize the presence of charachters broken or interfering with noise components, as described in the following.

We found that clean characters are characterized approximately by values of P^2/A in the range $(16, 150]$ (where P was computed as the number of background pixels 4-connected to at least one foreground pixel, and A is the number of foreground pixels) and of the aspect ratio (the ratio between width and height) in the range $[0.25, 2.5]$. Instead, the components corresponding to broken characters and to small background noise can exhibit a lower P^2/A value than 16, while components corresponding to two or more characters connected through noise components, can exhibit a larger P^2/A values than 150 or an aspect ratio outside the above interval. The above observations suggested us two measures of the degree of image noise, one related to the presence of broken characters or small noise components (denoted as f_1), the other to characters connected through large noise components (f_2). In both cases the starting point is a binarized image, on which all connected components are first identified (in this work, 8-connectivity is used to this aim), and the P^2/A value and aspect ratio are computed for each of them.

The f_1 and f_2 measures are based on the following rationale: if a text area contains the considered kind of noisy text, then small regions of the text area will contain on average both character-like and noise-like components. Accordingly, we first subdvided an image into a grid of $p \times q$ equally sized cells $\mathrm{C}_{ij}, i = 1, \ldots, p; j = 1, \ldots, q$ ($p = q = 10$ was used in this work), and then computed the average fraction of noise-like components over all cells. More precisely, f_1 was computed by considering for each cell C_{ij} the components whose center of mass belongs to C_{ij}. The number c_{ij} of character-like components (the ones with P^2/A in the range $(16, 150]$ and aspect ratio in the range $[0.25, 2.5]$), and the number n_{ij} of noise components (the ones with either P^2/A or aspect ratio, or both, outside the above intervals) was then computed. Then, considering only the cells with $c_{ij} > 0$, i.e. containing at least one character-like component (we denote the number of such cells with c_0^1), we computed for each of them the fraction of noise components $f_{ij} = n_{ij}/(n_{ij}+c_{ij}) \in [0, 1)$, where a vaue of 0 denotes the presence of clean text. We finally computed the average fraction of noise components over the c_0^1 cells, $f_1 = \frac{1}{c_0^1}\sum_{\mathrm{C}_{ij}:c_{ij}>0}\frac{n_{ij}}{n_{ij}+c_{ij}} \in [0, 1]$. If $c_{ij} = 0$ for all cells (namely, $c_0^1 = 0$), this means that the text area contains only noise-like components: f_1 is defined accordingly as 1.

The computation of f_2 is similar. However, since the considered noise components can spread over several cells, we count the number of pixels of character-like components (c^{P}_{ij}) and of noise-like components whose P^2/A value is higher than 150 (n^{P}_{ij}) lying into each cell C_{ij}. Then, analogously to f_1, we consider only cells containing at least one character-like component (let c_0^2 be their number), and compute the average fraction of noise components, $f_2 = \frac{1}{c_0^2}\sum_{\mathrm{C}_{ij}:c^{\mathrm{P}}_{ij}>0}\frac{n^{\mathrm{P}}_{ij}}{n^{\mathrm{P}}_{ij}+c^{\mathrm{P}}_{ij}} \in [0, 1]$, where a value of 0 denotes clean text. Accordingly, if $c^{\mathrm{P}}_{ij} = 0$ for all cells (namely, $c_0^2 = 0$), f_2 is defined as 1.

We finally devised a third measure (f_3) aimed at detecting the presence of large background noise components overlapping with characters, which can occur when text is placed over non-uniform background as in Figure 2 (center-bottom). To this aim, we extract the edges from the original image using the Canny algorithm, and compute the average number of edge pixels which lie *inside* each component of the binarized image. More precisely, the fraction of edge pixels which lie inside a component C is computed as the number of edge pixels which correspond to any pixel of C, excluding the pixels of its inner perimeter (defined as the set of pixels of the component 4-connected with at least one background pixel in the binarized image), divided by the total number of edge pixels that are not part of the inner perimeter of any component. The f_3 measure lies as well in the range $[0, 1]$, where 0 denotes the absence of the considered kind of noise.

In the next section we present some preliminary experiments to assess to what extent the measures defined above are able to detect the presence and measure the amount of the considered kind of image noise.

4. EXPERIMENTAL RESULTS

We give first a demonstration of the capability of the three proposed measures to detect and measure the extent of the considered kinds of noisy text. To this aim we constructed an image which contains the 26 characters of the English alphabet, both uppercase and lowercase (see Figure 1). We considered four different obscuring techniques shown in Figure 1 (two examples with different amounts of noise are shown for each technique): from top-left to bottom-right: small background noise components around text; a grid made up of 1-pixel wide lines of the same colour as the background overlapped with text; a non-uniform background; a grid made up of 1-pixel wide lines of the same colour as the text, overlapped with it. Obscuring techniques similar to the first three above were observed by the authors in real spam images (see Figure 2), while the last one was proposed in the EZ-Gimpy visual CAPTCHA used by Yahoo [8]. The values of the f_1, f_2 and f_3 measures turned out to be 0 for the clean image, correctly denoting the presence of clean text. The values of the three measures for the noisy images reported in Figure 1 are shown below each image. The two obscuring techniques in the four top images of Figure 1 resulted in broken characters or in small background noise components around characters, which are the focus of the f_1 measure. As expected, it can be seen that the f_1 values increase as the degradation level increases, while the values of f_2 and f_3 (which are aimed to detect different effects of obscuring techniques) remain zero. The same happens to the f_3 measure for the first and second image (bottom), when large background components turn out to hide some characters, and to the f_2 measure when characters in the binarized image turn out to be connected through noise components (the two right-bottom images).

We now report some preliminary experimental results on a data set of 186 real spam images collected at the personal authors' mailboxes (since no publicly available data sets of spam images is available yet, to our knowledge), available at `http://ce.diee.unica.it/spam-images.zip`. Image binarization was carried out using the demo version of the commercial software ABBYY FineReader 7.0 Professional,[3] with default parameter settings. Content obscuring techniques aimed at defeating OCR tools were applied by spammers on 96 images (see Figure 2). Note that some of the obscuring techniques did not result in the kinds of noise considered in Section 3 (as in Figure 2, bottom-right image). We found that on 29 out of the 96 noisy images no noise remained after binarization. The remaining 90 images were clean or contained a limited amount of random noise probably aimed at defeating detection techniques based on digital signatures instead of OCR tools.

Figure 3 shows the values of the three measures for each of the 186 spam images. Most of the 119 clean binarized images turned out to exhibit low values of f_1, f_2 and f_3. Instead, the 67 noisy binarized images are spread across a larger range of values for the three measures. In particular, we observed that each measure took on high values for images characterized by the presence of the corresponding kind of noise. As an example, consider the third image from top in Figure 2: in the binarized image some characters are broken or connected by background line segments, but no one is hidden by background components. This is revealed by high values of f_1 and f_2, respectivey 0.58 and 0.89, while the value of f_3 is low (0.08) as expected. Consider finally the case of spam images in which the effect of noise is different than the one addressed by these three measures, as in the bottom-right image of Figure 2: in the binarized image there is no significant amount of character breaking or merging, and there are no background noise components. The values of three measures turned out to be low as expected ($f_1 = 0.08$, $f_2 = f_3 = 0$). As explained in Section 3, to detect such kind of obscuring techniques other measures should be used, for instance related to character alignment or deformation.

To sum up, the above preliminary experiments provided some evidence that our approach can be exploited to design modules of spam filters aimed at detecting image spam as described in section 2, and in particular the use of content obscuring techniques applied to text embedded into images, which is one of the main charateristics of current image spam and it is likely to be widely used in the near feature. We believe that works in the OCR literature, in particular related to image quality measures, and recent works on CAPTCHAs could give further useful suggestions for the development of this research direction. Ongoing work is aimed at evaluating the proposed approach on legitimate images.

5. REFERENCES

[1] A. Androutsopoulos, J. Koutsias, K. V. Cbandrinos, and C. D. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proc. ACM Int. Conf. on Research and Developments in Information Retrieval*, pages 160–167, 2000.

[2] H. Aradhye, G. Myers, and J. A. Herson. Image analysis for efficient categorization of image-based spam e-mail. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 914–918, 2005.

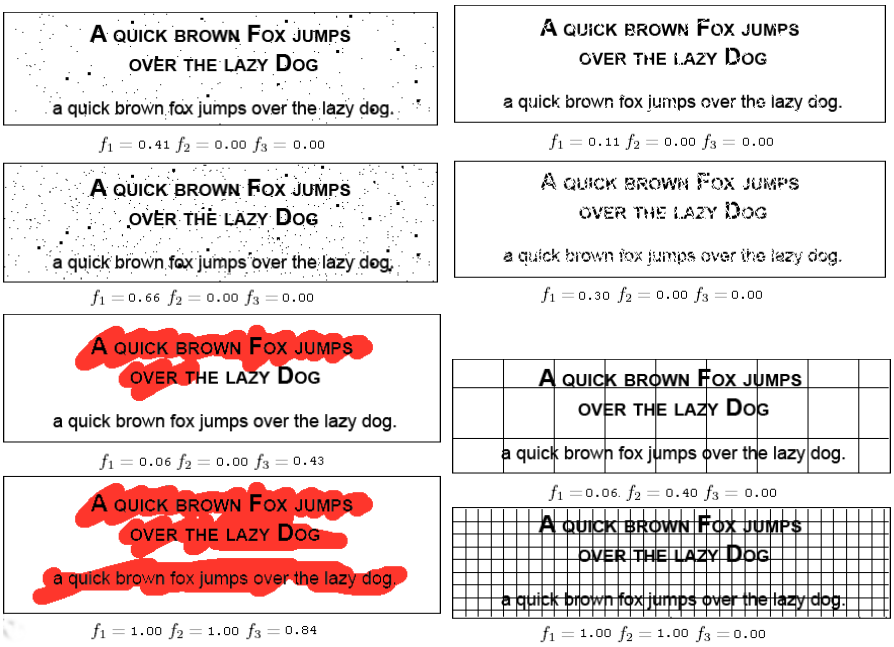
[3] H. S. Baird and M. Chew. Baffletext: a human interactive proof. In *Proc. IS&T/SPIE Document Recognition & Retrieval Conf.*, 2003.

[4] L. R. Blando, J. Kanai, and T. A. Nartker. Prediction of OCR accuracy using simple image features. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 319–322, 1995.

[5] H. Drucker, D. Wu, and V. N. Vapnik. Support vector machines for spam categorization. *IEEE Transaction on Neural Networks*, 10(5):1048–1054, 1999.

[6] G. Fumera, I. Pillai, and F. Roli. Spam filtering based on the analysis of text information embedded into

[3] `http://www.abbyy.com`



$f_1 = 0.41$ $f_2 = 0.00$ $f_3 = 0.00$

$f_1 = 0.11$ $f_2 = 0.00$ $f_3 = 0.00$

$f_1 = 0.66$ $f_2 = 0.00$ $f_3 = 0.00$

$f_1 = 0.30$ $f_2 = 0.00$ $f_3 = 0.00$

$f_1 = 0.06$ $f_2 = 0.00$ $f_3 = 0.43$

$f_1 = 0.06$ $f_2 = 0.40$ $f_3 = 0.00$

$f_1 = 1.00$ $f_2 = 1.00$ $f_3 = 0.84$

$f_1 = 1.00$ $f_2 = 1.00$ $f_3 = 0.00$

Figure 1: Four obscuring techniques applied to an artificial image with embedded text (two different "amounts" of noise are shown for each technique).

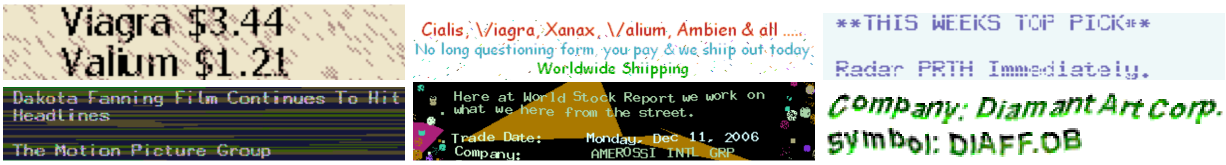

Viagra $3.44
Valium $1.21
Cialis, \/iagra, Xanax, \/alium, Ambien & all .....
No long questioning form, you pay & we ship out today
Worldwide Shipping
THIS WEEKS TOP PICK
Radar PRTH Immediately.
Dakota Fanning Film Continues To Hit Headlines
The Motion Picture Group
Here at World Stock Report we work on what we here from the street.
Trade Date: Monday, Dec 11, 2006
Company: AMEROSSI INTL GRP
Company: DiamantArt Corp.
symbol: DIAFF.OB

Figure 2: Examples of real spam images (details) from the data set used for the experiments.

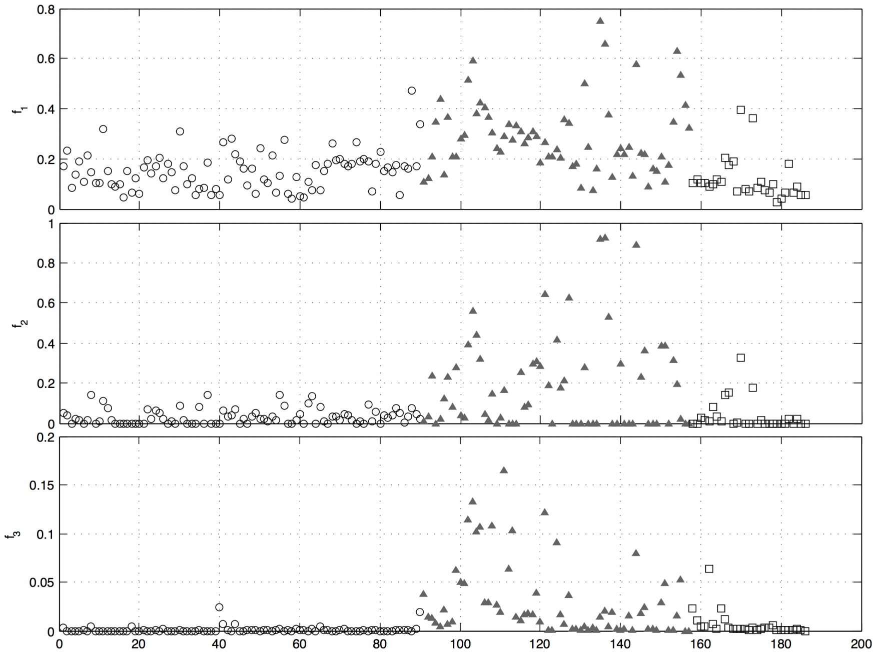


Figure 3: Plots of the f_1, f_2 and f_3 measures for the 186 spam images (ordered along the x-axis). ○: clean images; □: obscured images (clean binarized images); △: obscured images (noisy binarized images).

images. *Journal of Machine Learning Research (special issue on Machine Learning in Computer Security)*, 7:2699–2720, 2006.

[7] P. Graham. A plan for spam, 2002. http://paulgraham.com/spam.html.

[8] G. Mori and J. Malik. Recognizing objects in adversarial clutter: breaking a visual captcha. In *Proc. Int. Conf. Computer Vision and Pattern Recognition*, volume I, pages 134–141, 2003.

[9] D. G. Pelli, C. W. Burns, B. Farell, and D. C. Moore-Page. Feature detection and letter identification. *Vision Research*, 46:4646–4674, 2006.

[10] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. *AAAI Technical Report WS-98-05, Madison, Wisconsin*, 1998.

[11] C.-T. Wu, K.-T. Cheng, Q. Zhu, and Y.-L. Wu. Using visual features for anti-spam filtering. In *Proc. IEEE Int. Conf. on Image Processing*, volume III, pages 501–504, 2005.