Visual and OCR-Based Features for Detecting Image Spam

Francesco Gargiulo and Carlo Sansone

Dipartimento di Informatica e Sistemistica, University of Naples Federico II
Via Claudio 21, I-80125, Naples, Italy

Abstract. The presence of unsolicited bulk emails, commonly known as *spam*, can seriously compromise normal user activities, forcing them to navigate through mailboxes to find the - relatively few - interesting emails. Even if a quite huge variety of spam filters has been developed until now, this problem is far to be resolved since spammers continuously modify their malicious techniques in order to bypass filters. In particular, in the last years spammers have begun vehiculating unsolicited commercial messages by means of images attached to emails whose textual part appears perfectly legitimate.
In this paper we present a method for overcoming some of the problems that still remain with state-of-the-art spam filters when checking images attached to emails. Results on both personal and publicly available email databases are presented, in order to assess the performance of the proposed approach.

1 Introduction

It is well known that unsolicited bulk emails, commonly known as *spam*, are a serious problem for email accounts of single users, small companies and large institutions, since the presence of spam can seriously compromise normal user activities, forcing them to waste time, bandwidth and storage space. Moreover, spam emails have often unsuitable content (as a pornographic material advertising) that could be illegal for minors.

In this realm, different counter-measures to spam have been proposed, using *regulatory* or *technical* approaches. The legislative approach did not obtain the desired results. Several technical approaches have thus been implemented in different anti-spam filters currently used to detect unsolicited bulk emails [4, 12].

In the past, researchers have first addressed this problem as a text classification or categorization problem [1, 5]. However, as spammers' techniques continue to evolve and the genre of email content becomes more and more diverse, keywords-based anti-spam approaches alone are no longer sufficient. Then, different techniques have been used to analyze the mail text, the majority of which are learning-based approaches. Considering the spam detection as a binary classification problem, several algorithms from learning theory field can be used, such as Bayesian algorithms [11] or Support Vector Machines (*SVM*) [7]. These systems, using the acquired knowledge on a suitable training set, are able to discriminate between legitimate and malicious text in order to reject mails considered as spam.

Gargiulo F. and Sansone C. (2008).
Visual and OCR-Based Features for Detecting Image Spam.
In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems*, pages 154-163

Most of the previous approaches use also feature extraction techniques. Features extracted from email's text are then given as input to a classifier in order to filter spam messages from legitimate texts. Anyway, spammers adopted different solutions to mislead this kind of filters by obscuring text, by obfuscating words with symbols and by including neutral text to confuse the classification process. These tricks have been studied by anti-spam researchers in order to find new solutions to restore filtering effectiveness.

Among the different tricks used by spammers, an emerging kind of spam practice is the so-called *image spam*. Here spammers use to sent their messages in attached images that are readable by human but hidden from the filter. Even if image spam is relatively new, various proposals have been made in the literature to address this kind of spam, too. Most approaches use some form of embedded text detection within images. The rationale is that spam images should contain a text whose content can spread unsolicited commercial messages.

In particular, Wu et al. [13] defined a set of visual features in order to detect characteristics common in spam images, such as embedded text and banner features. These features are then combined with message text features for training a one-class SVM that should be able to detect when legitimate (*ham*) emails are outside the spam class. Similarly, Aradhye et al. [2] proposed features to detect embedded text and some background types that should be consistent with spam. Once again, they use an SVM classifier to discriminate between ham and spam images. A different approach is instead followed in [8]. Here the authors propose to process attached images with a state-of-the-art OCR and then to forward OCR outputs to a text-based spam filter.

All the aforementioned approaches, however, cannot be used when text within images is voluntarily distorted and/or obfuscated. As it was noted in [3], in fact, now spammers try to make OCR and text detection techniques ineffective without compromising human readability, by placing text on non-uniform background, or by using techniques like the ones exploited in CAPTCHAs[1] (programs that generate and grade tests that humans can pass but current computer programs cannot).

In a recent paper, Dredze et al. [6] presented an approach to image spam detection based on an algorithm for speed sensitive feature selection. Despite the focus of the paper is mainly on a method that can efficiently process attached images, it is interesting to note that their approach consider both feature that relies on metadata and other simple image properties (such as size and format) as well as features related to the visual content of the image itself. So, they neither try to detect text within images, nor consider the fact that now spammers use tricks for obfuscating this text.

In this paper we define a method for overcoming some problems that still exist with state-of-the-art spam filters when addressing image spam. In particular, we tried to fuse the key ideas of some of the previously described approaches, by defining two different sets of features. A first set should characterize an image from a global point of view, in order to detect artifacts that are typically indications of the presence of spam. Another set of features has been instead devised for detecting malicious text in images,

[1] The term CAPTCHA (Completely Automated Turing Test To Tell Computers and Humans Apart) was coined in 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper and John Langford of Carnegie Mellon University. At that time, they developed the first CAPTCHA to be used by Yahoo – http://www.captcha.net/

by explicitly taking into account the fact that now such text is typically deformed and/or obfuscated by spammers. In order to do that, the output of an OCR applied to the image under test is further processed for deriving features that should be able to characterize this kind of trick.

The rest of the paper is organized as follows. First, we present the proposed method for coping with image spam, by giving details about the proposed features. Then, the databases used for assessing the performance of our method are introduced and experimental results are presented and discussed. A comparison with other state-of-the-art image spam filter is also reported. Finally, some conclusions are drawn.

2 The Proposed Approach

As stated in the introduction, nowadays various solutions are proposed by the anti-spam community for the detection of image spam. A conservative approach for eliminating image spam can be realized by blocking images from unknown senders, or even by blocking images from all the senders. Obviously, several legitimate emails will never get their destination. Apart from these extreme methods, a certain number of research systems addressed this problem. Most of them use simple visual features to distinguish between ham and spam images. Other papers focused on the possibility to use a full Optical Character Recognition (OCR) and then to apply to its output the same techniques adopted for processing the textual part of an email. These systems, however, suffer from the problem that images are frequently adulterated and so the OCR output cannot be correctly processed by means of a textual analysis, as you can see in Figure 1 (right). Moreover, they cannot address images without embedded text.

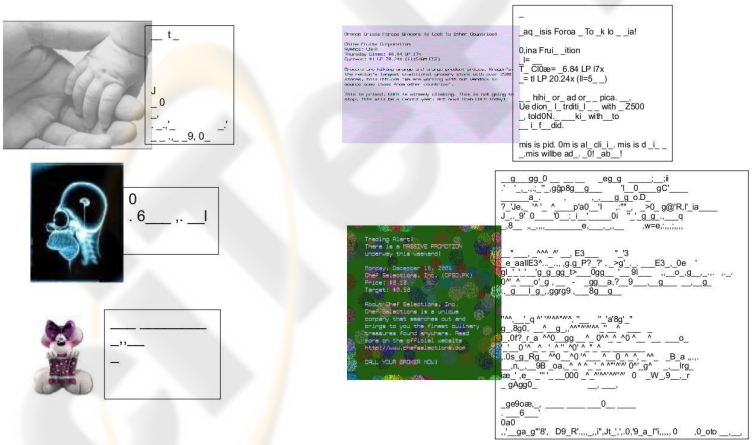

Fig. 1. Outputs obtained by applying *gocr* (available at http://jocr.sourceforge.net) to some ham (left) and spam (right) images.

Starting from these considerations, in this paper we propose a novel approach for the detection of the image spam in which two different image processing techniques (see

Figure 2) are used. The first one is devoted to directly extract some global features from each image attached to the emails. Such features should be able to detect if images were adulterated or not, by considering the complexity of the image itself as it is perceived from an human being. The second processing is carried out by means of two steps. First, there is a preprocessing phase in which an OCR is used; then, a feature extraction process try to characterize the OCR output, in order to detect if it contains embedded text that has been voluntarily obfuscated and/or distorted. The differences between our use of an OCR and the previous ones (see for example [8]) is that in our case the OCR was used as a tool for obtaining a fingerprint of an adulterated image, which is then characterized by means of a suitable set of features.

The features directly extracted from the image and those obtained from the OCR output are then put together as input of a binary classifier that, after a suitable training phase, can decide if the image under test is ham or spam (see Figure 2).

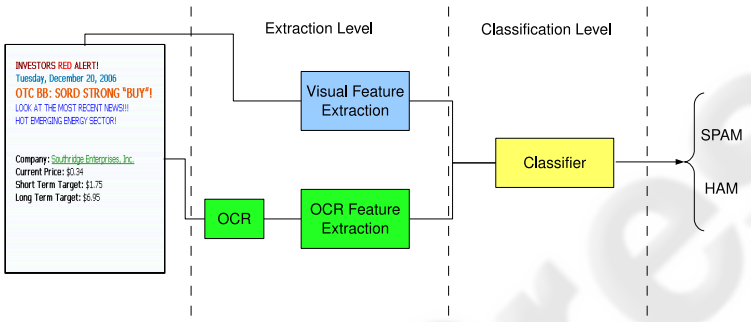

Fig. 2. The proposed approach for filtering image spam.

If there is more than one image attached to an email, we perform a boolean OR among the output of the different classification acts, i.e. an email is declared as spam if there is at least one attached image recognized as spam. We do not use any sort of voting mechanism in order to deny a kind of *padding-attack* from the spammers. That is, the possibility that an attacker puts a spam message within a *normal* context, by attaching various images only one of them vehiculates the spam message. In this case, if a majority voting was used, the system could erroneously assign the mail to the ham class.

It is worth noting that the proposed approach can also be integrated in a more general architecture, so as the one presented in [9], which has been devised to cope with both image spam and text-based spam. In that proposal, the output of an OCR applied to each image is also forwarded as-is to a text-based filter so as to have an additional verdict about the fact that the image under test could be spam. Since the OCR output could be not processable by a text-based filter, an additional module is however needed for deciding when such an output must be passed to the text-based filter.

2.1 Visual Features

The first set of features, that we called *visual features*, are directly obtained from the image attached to the mails. In order to give an image characterization that should be

able to discriminate between normal and adulterated images, we considered features that describe the image texture from a statistic point of view. As said before, in fact, spammers typically now try to bypass filters that use an OCR for detecting texts within an image by obfuscating such texts with the addition of some noise or by superimposing a texture (see also Figure 1–right). So, texture detection can help in individuating images that contain spam messages. For the sake of simplicity, in the following we will present the considered features in case of gray-level images, but the same operators can be applied to color images too.

We will use $\{I(x,y), 0 \leq x \leq N-1, 0 \leq y \leq M-1\}$ to denote an $N \times M$ image with G gray levels. All the considered statistical texture measures are based on the co-occurrence matrices. Spatial gray level co-occurrence estimates image properties related to second-order statistics. The $G \times G$ gray level co-occurrence matrix $P_{\mathbf{d}}$ for a displacement vector $d = (dx, dy)$ is defined as follows. The entry (i,j) of $P_{\mathbf{d}}$ is the number of occurrences of the pair of gray levels i and j which are a distance $\mathbf{d}$ apart. Formally, it is given as:

$$P_{\mathbf{d}}(i,j) = |\{((r,s),(t,v)) : I(r,s) = i, I(t,v) = j\}|$$

where $(r,s),(t,v) \in N \times M$, $(t,v) = (r+dx, s+dy)$, and $|.|$ is the cardinality of a set.

As regards the choice of the displacement vector $\mathbf{d}$, we considered the four direct neighbors of each pixel, i.e. we used four pairs as values of dx and dy for calculating the number of co-occurrences, namely $(0,1)$, $(1,0)$, $(-1,0)$ and $(0,-1)$. We do not perform a normalization of $P_{\mathbf{d}}$ in order to preserve the dependence of the considered features on the image size.

As suggested in [10], from the co-occurrence matrix it is possible to extract features that can be used for detecting a texture within an image. In particular, we considered the following five features:

– **Contrast:**

$$\sum_i \sum_j (i-j)^2 P_{\mathbf{d}}(i,j)$$

is the difference in terms of visual properties that makes an object (or its representation within an image) distinguishable from other objects and the background. In the visual perception of real world, contrast is determined by the difference in the color and brightness of the object and other objects within the same field of view. In practice, it is the ratio between the brightest and the darkest value of the image. In the case of a B/W image, note that the increase of the contrast is equal to erase gray values.

– **Entropy:**

$$-\sum_i \sum_j P_{\mathbf{d}}(i,j) log P_{\mathbf{d}}(i,j)$$

is an index of the brightness variation among the pixel in an image. More the values of brightness are different each others, more the entropy will be higher.

– **Energy:**

$$\sum_i \sum_j P_{\mathbf{d}}^2(i,j)$$

is the spectral content of an image

– **Correlation:**

$$\frac{\sum_i \sum_j (i-\mu_x)(j-\mu_y) P_{\mathbf{d}}(i,j)}{\sigma_x \sigma_y}$$

is an index of the correlation degree among the pixel. Here μ_x and μ_y are the means and σ_x and σ_y are the standard deviations of $P_{\mathbf{d}}(x)$ and $P_{\mathbf{d}}(y)$ respectively, where $P_{\mathbf{d}}(x) = \sum_j P_{\mathbf{d}}(x,j)$ and $P_{\mathbf{d}}(y) = \sum_i P_{\mathbf{d}}(i,y)$

– **Homogeneity:**

$$\sum_i \sum_j \frac{P_{\mathbf{d}}(i,j)}{1+|i-j|}$$

is a measure of the brightness variation within the image. If the image is completely black or white, its homogeneity value will be the maximum. On the contrary, if the image contains several brightness variations, this value will be very low.

Another category of features that can be used for characterizing images from a global point of view is based on the complexity of an image for a human reader. We have chosen to consider a feature also proposed in [3]:

– **Perimetric Complexity:** is defined as the squared length of the boundary between black and white pixels (the perimeter) in the whole image, divided by the black area.

Note that, differently from [3], we evaluate the perimetric complexity on the whole image, after performing a binarization with a fixed threshold.

2.2 OCR-based Features

As it can be seen in Figure 1, when an OCR is used for processing images whose embedded texts have been distorted or obfuscated, the majority of the words cannot be correctly detected. Furthermore, several characters that typically are not present in common-sense words can appear in the OCR output. So, we defined some OCR-based features for obtaining a characterization of this kind of text. The features we are investigating on are mainly based on the presence of *special* characters, i.e. those characters that should not be frequently present in a legitimate text. The whole set we considered is made up of the following characters: { **!**, **”**, **#**, **$**, **%**, **&**, **’**, **(**, **)**, *****, **+**, **,**, **-**, **. . .**, **/**, **@**, **ˆ**}.

Starting from this set we defined six OCR-based features:

– **text_length:** the number of characters of the whole text extracted by the OCR
– **words_number:** the number of words in the text extracted by the OCR
– **ambiguity:** the ratio between the number of special and normal characters

– **correctness:** the ratio between the number of words that do not contain special characters and the number of words that contain special characters
– **special_length:** the maximum length of a continuous sequence of special characters
– **special_distance:** the maximum distance between two special characters, i.e., the longest sequence of normal characters between two special characters.

3 Experimental Results

In the following we will first present the two databases used for evaluating the effectiveness of the proposed approach. Then, we will evaluate if the use of both visual and OCR-based features can improve the performance of the system with respect to the use of a single set of features. Finally, we present a comparison of our approach with a state-of-the-art anti-spam filter, i.e. *SpamAssassin* equipped with two different spam image plug-ins, on a personal corpus of emails with attached images. Moreover, we also make a comparison of our approach with the one presented in [6] on a set of publicly available images. We do not compare our results with the ones obtainable with *SpamAssassin* on the latter dataset since it is made up of images only, without the original emails.

As regards the two datasets, whose details are given in Table 1, the first one (that we have called *UNINA*) is composed by 3395 emails with attached images. Emails were collected from the mailboxes of few users of the `studenti.unina.it` mailserver in a period of about three years (2005-2007). This mailserver hosts the mailboxes of all the students of the University of Naples Federico II. Among those emails, 151 contain ham images and 3244 contain spam images. A subset of these images is shown in Figure 3.

Fig. 3. Same ham (left) and spam (right) images taken from the UNINA dataset.

The second dataset (hereinafter denoted as *DREDZE*) was presented in a paper by Dredze et al. [6] and was publicly available[2]. It was made up of 5306 images (2008 ham and 3298 spam) collected from 10 email accounts across 10 domains and a catch-all filter on two domains over the period of one month. Every attached image (*gif*, *jpg*,

[2] http://www.cis.upenn.edu/ mdredze/datasets/image_spam/

png and *bmp*) was extracted from that emails, excluding images smaller than 10x10 pixels since these are often used as blank spacers in HTML documents.

Table 1. Details about the datasets used in our tests.

Dataset	*Spam Images*	*Ham Images*
UNINA	3244	151
DREDZE	3298	2008

In order to make a fair comparison with the results presented by Dredze et al. in [6], we adopt for the classification stage of our approach one of the classifiers used in their work, i.e. a *Decision Tree Classifier*. In particular, a C4.5 (J48) coming from the open source tool *Weka*[3] was selected. Moreover, it is worth noting that in all the test reported hereinafter our results are given in terms of the average accuracy obtained by means of a *10-fold* cross validation.

In Table 2 the results obtained on the *UNINA* dataset by our approach are presented. In particular, it can be noted that the adoption of both visual and OCR-based features improves the performance obtainable by the system with respect to the case in which only visual or OCR-based features are used.

Table 2. Results obtained on the *UNINA* dataset with different feature sets.

Features	*Accuracy*
Visual	94.31%
OCR-Based	94.79%
Both	96.98%

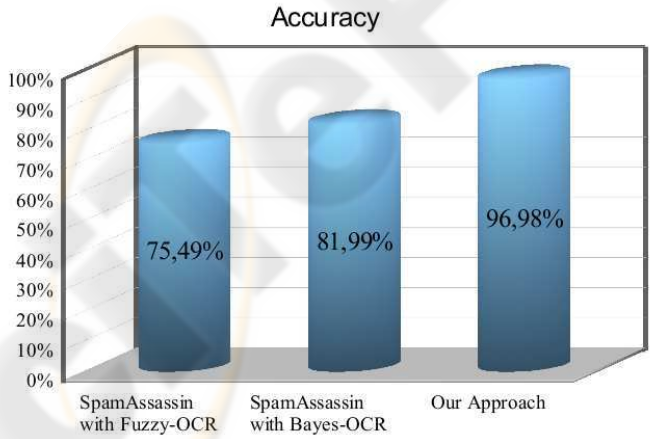

Fig. 4. Comparison between the proposed approach and *SpamAssassin* with *Bayes-OCR* and *Fuzzy-OCR* on the *UNINA* dataset.

[3] http://www.cs.waikato.ac.nz/ml/weka/

In Figure 4 we report a comparison of the results obtained by our system on the emails of the *UNINA* dataset with those obtainable with *SpamAssassin* equipped with two plug-ins devised for filtering image spam, namely *Bayes-OCR*[4] and *Fuzzy-OCR*. The standard *SpamAssassin* configuration was used. As it is evident, our approach significantly outperforms both *Bayes-OCR* and *Fuzzy-OCR*. However, it has to be remarked that, differently from our method, *SpamAssassin* takes a decision by also considering the body of the email, if it is present.

Finally, in Table 3 the comparison between our approach and the one presented in [6] is shown. Results obtained by [6] and reported here refer to the case in which the whole set of the features they proposed have been processed by using a Decision Tree classifier. In this case, the *F1* measure is also shown, since this figure of merit is used in [6] too. These results confirmed the effectiveness of our approach, which performs slightly better than the system described in [6].

Table 3. Comparison between the proposed approach and the one presented in [6] on the *DREDZE* dataset.

	Accuracy	*F1*
Our approach	97%	0.97
Dredze et al.	96%	0.96

4 Conclusions

In this paper we presented an approach for coping with spam images that contain embedded texts voluntarily deformed and/or obfuscated by spammers. The effectiveness of the proposed approach was demonstrated on two different datasets of images collected from real emails.

As future work, we plan to integrate the proposed method in a more general architecture that could also be able to address spam sent via pure textual emails. Moreover, it should be also interesting to study the possibility of using a set of OCRs as preprocessors before extracting the OCR-based features. In that case a suitable methodology for combining different ORC outputs should be provided, too. Finally, we want to better investigate the robustness of the approach when dealing with legitimate images, such as low-quality scanned documents, which contain a complex text that cannot be easily processed by an OCR.

References

1. Androutsopoulos, I., Koutsias, J., Chandrinos, K., Paliouas, G., Vassilakis, C.: An Evaluation of Naive Bayesian Anti-Spam Filtering. In: Potamias G., Moustakis V., van Sommeren M. (eds): Proceedings of the 11th European Conference on Machine Learning (2000) 9–17

[4] This plug-in is available for download at the URL: http://prag.diee.unica.it/n3ws1t0/?q=node/107

2. Aradhye, H.B., Myers, G.K., Herson, J.A.: Image Analysis for Efficient Categorization of Image-based Spam E-mail. Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR '05). IEEE Computer Society Press, Washington DC USA (2005) 914–918
3. Biggio, B., Fumera, G., Pillai, I., Roli, F.: Image Spam Filtering Using Visual Information. Proceedings of the 14th International Conference on Image Analysis and Processing (ICIAP '07). IEEE Computer Society Press, Washington DC USA (2007) 105–110
4. Blanzieri, E., Bryl, A.: A Survey of Anti-Spam Technique. Technical Report DIT-06-056, Informatica e Telecomunicazioni, University of Trento (2006)
5. Cohen, W.: Learning rules that classify e-mail. AAAI Spring Syposium on Machine Learning in Information Access (1996) 18–25
6. Dredze, M., Gevaryahu, R., Elias-Bachrach, A.: Learning Fast Classifiers for Image Spam. Proceedings of the Third Conference on Email and Anti-Spam (2007)
7. Drucker, H., Wu, D., Vapnik, V.N.: Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks 10(5) (1999) 1048–1054
8. Fumera, G., Pillai, I., Roli, F.: Spam filtering based on the analysis of text information embedded into images. Journal of Machine Learning Research, 7 (2006) 2699–2720
9. Gargiulo, F., Penta, A., Picariello, A., Sansone, C.: Using visual and semantic features for Anti-Spam filters. Workshop on Machine Learning in Adversarial Environments for Computer Security, NIPS 2007, Whistler BC Canada (2007)
10. Haralick, R.M.: Statistical and Structural Approaches to Texture. Proceedings of the IEEE, 67 (5) (1979) 786–804
11. Metsis, V., Androutsopoulos, I., Paliouras, G.: Spam Filtering with Naive Bayes - Which Naive Bayes?. Proceedings of the Second Conference on Email and Anti-Spam, Mountain View CA USA (2006)
12. Schryen, G.: Anti-Spam Measures, Analysis and Design. Springer (2007)
13. Wu, C.-T., Cheng, K.-T., Zhu, Q., Wu, Y.-L.: Using Visual Features for Anti-Spam Filtering. Proceedings of the IEEE International Conference on Image Processing, September 11-14, Genova Italy (2005)