ABSTRACT

We tackle the problem of the steganalysis of images produced by different sources. We first use a classifier to try to understand the image source and then use a version of the steganalyzer that has been explicitly trained to work with images belonging to the correct class. These classifiers are widely available from the image forensics literature and have reached a good level of maturity hence making the proposed approach feasible. We tested the goodness of our approach on a case study in which a steganalyzer is asked to analyze both computer generated and camera images. The results we obtained are promising encouraging further research in this direction.

Keywords: Steganalysis, Image forensics, JPEG steganography.

1. INTRODUCTION

It is well known that the performance of a steganalyzer depends on the knowledge that the steganalyzer has on the steganographic algorithm. In this sense, a distinction is often made between targeted steganalysis, in which knowledge of the underlying steganographic algorithm is assumed, and blind steganalysis, in which the details of the steganographic algorithm are assumed to be unknown [1, 2]. Another kind of knowledge that would be extremely helpful for the steganalyzer regards the statistics of the cover images. In most cases, in fact, it is essential that the steganalyzer is trained on a set whose characteristics are representative of the statistics of the images the steganalyzer will have to work with (usually modeled by a properly designed test set). This problem is often neglected or underestimated in the current literature, yet a mismatch between the statistics of the training set and those of the test set can significantly decrease the performance of a steganalyzer [3]. All the more that in practical applications it is likely that the steganalyzer does not have access to precise information about the statistics of the cover images. If this is the case, how should the steganalyzer be built? A common approach is to estimate the statistics of the cover images without having them is through calibration, a procedure whereby the steganalyzer tries to recover the statistics of the cover image by attacking the analyzed image. Possible approaches include image resizing [4] or desynchronization by block-by-block compression [1]. In both cases, the features used for steganalysis are extracted from the attacked image and used to normalize the features extracted from the previously non-attacked image. Another widely adopted way to steganalyze images coming from different sources is to train the steganalyzer on a set that is as heterogeneous as possible, hoping that in this way the steganalyzer will learn how to recognize the presence of the stegomessage regardless of the statistics of the analyzed images. In this way, however, the statistics of the cover images will tend to be too broad, hence making it more difficult to distinguish them from stego-images.

In this paper, we consider an alternative approach that works whenever the images belong to a number of different classes known beforehand. The basic idea is to let the steganalyzer be preceded by a classification step during which the image is classified as belonging to one of the possible classes. Then the image is analyzed by a steganalyzer that has been trained on a proper training set. By implementing the above idea in a simple case in which the images may be either computer generated or acquired by a digital camera, we show that the considered approach may outperform significantly a steganalyzer that is trained on a mixed dataset and used to classify the images belonging to the two classes.

The rest of the paper is organized as follows. In section 2, we describe the proposed system, in section 3 we give the details of the case study we used to validate our idea, while in section 4 we present the results of the experiments. The paper ends in section 5 where we give our conclusions.

2. THE PROPOSED SCHEME

Our system is based on a very simple idea: instead of trying to build a steganalyzer that is capable of handling images belonging to different classes, first decide which class the image at hand belongs to and then use a steganalyzer explicitly designed to work with images of that class. In the following we will refer to a steganalyzer built according to the above idea as a forensics-aided steganalyzer (FA-steganalyzer), since in the case study considered in this paper the pre-classification step is nothing but a forensics step in charge of understanding the provenance of the image at hand. Alternatively we could refer to it as a switch-steganalyzer, since it switches between different steganalyzers depending on the characteristics of the to-be-analyzed image. The overall architecture of the FA-steganalyzer is depicted in figures 1 and 2. In the training phase (figure 1) several versions of the same steganalyzer trained on images belonging to the classes $C_1, C_2 \ldots C_M$ are built. Let us call such steganalyzers $S_1, S_2 \ldots S_M$. At the same time a classifier $C$ is trained to distinguish between images belonging to the various classes. The actual steganalysis is carried out by first classifying the image at hand as belonging to one of the possible classes, and then using the version of the steganalyzer that was trained on the correct class of images (figure 2).

2.1. Error probability

The performance of the FA-steganalyzer depends on several factors, including the error probability of the various steganalyzers when they are applied to images coming from the various classes, and the error probability of the forensics classifier. To elaborate let $P_e^S / (I \in C_i)$
\[ C_i = P^s_{S_j}(i) \] be the probability that the steganalyzer \( S_j \) makes an error when is asked to classify an image \( I \) belonging to \( C_i \). Let then \( P^c(j|I \in C_i) = P^c(j|i) \), be the probability that the classifier classifies an image coming from \( C_i \) as belonging to \( C_j \). Finally, let \( P(C_i) \) be the probability that the selector in Fig. 2 picks up an image from the \( i \)-th class. The overall error probability of the FA-steganalyzer is:

\[ P_e = \sum_{i=1}^{M} \sum_{j=1}^{M} P^s_{S_j}(i) P^c(j|i) P(C_i). \tag{1} \]

In the following we will consider a scenario in which \( M = 2 \), i.e. only two image classes are possible. In this case, and by assuming that the a priori probabilities of the two classes are equal, the above equation assumes the following simpler form:

\[ P_e = \frac{1}{2} \left( P^s_{S_1}(1) P^c(1|1) + P^s_{S_1}(2) P^c(1|2) \right) + \frac{1}{2} \left( P^s_{S_2}(1) P^c(2|1) + P^s_{S_2}(2) P^c(2|2) \right). \tag{2} \]

If the classifier in charge of distinguishing between images coming from different sources is balanced, i.e. it has the same error probability regardless of the class the image belongs to, we have

\[ P^c(2|1) = P^c(1|2), \] yielding:

\[ P_e = \frac{1}{2} \left( P^s_{S_1}(1) + P^s_{S_2}(2) \right) P^c + \frac{1}{2} \left( P^s_{S_1}(1) + P^s_{S_2}(2) \right) P^c, \tag{3} \]

where by \( P^c \) and \( P^c \) we have indicated, respectively, the probability that the classifier makes a correct or a wrong decision.

Since we can assume that \( P^s_{S_1}(1) < P^s_{S_2}(1) \) and \( P^s_{S_2}(2) < P^s_{S_1}(2) \), it is evident that the higher the error rate of the classifier the worse the performance of the FA-steganalyzer.

The performance of the FA-steganalyzer should be compared against those of a scheme wherein the steganalyzer is trained on a mixed dataset and used to steganalyze all the images regardless of the class they belong to (in the following we refer to a steganalyzer built in this way as a mixed-steganalyzer). We could also consider a more straightforward solution in which a steganalyzer trained on images belonging to \( C_1 \) (or \( C_2 \)) is used to steganalyze also the images belonging to the other classes. This is equivalent to letting \( P^c(1|2) = P^c(1|1) = 0 \) (or equivalently \( P^c(2|2) = P^c(2|1) = 0 \)) in equation 2. It is easy to verify, that for reasonable values of the other parameters in equation (2), the error probability of such a steganalyzer would always be higher than that of the FA-steganalyzer.

### A CASE STUDY

In order to assess the validity of the FA-steganalyzer, the probabilities involved in equation (1) must be evaluated experimentally and the overall error probability compared to that of a steganalyzer that has been trained on a mixed dataset. In the following, the result of such a comparison is reported for a simple case study. To be specific, we consider a scenario in which the images may belong to two different classes, namely computer generated images and natural images acquired by a digital camera. The reason for such a choice is that the statistical differences between these two classes of images are likely to be significant ones, hence highlighting the problems inherent in the steganalysis of images stemming from heterogenous sources.

#### 3.1. The datasets

The scenario outlined above has been instantiated by considering two pairs of datasets. The first dataset pairs, let us call it \( D_1 \), is rather large since it includes 6000 computer generated (CG) images and 6000 camera images. CG images are very heterogeneous, since they include photo-realistic images, as well as cartoons and other easily distinguishable images. The camera images were taken from several common digital cameras like Canon, Nikon, Olympus and Sony. The images were in RAW format and we built the camera database by cropping and JPEG-compressing them with quality factor 90%. The size of the images range from 512 x 448 pixels to 512 x 512 pixel. We expect that the forensics analysis of the these two sets of images is rather easy, leading to good performance of the FA-steganalyzer. The second dataset (say \( D_2 \)) includes 764 CG and 764 natural images stemming from the database in [5]. The images in the dataset have a size ranging from 276 x 412 to 1196 x 762. The CG images in \( D_2 \) are very realistic ones making it difficult the distinction between CG and camera images. We expect that the FA-steganalyzer will encounter more difficulties with dataset \( D_2 \).

In all cases the images are available in JPEG format, with quality factors ranging from 70% to 100%, most of the images being compressed with a quality factor equal to 90%. The reason to work
with JPEG images is that they are by far the most common kind of images and hence any practical steganographic algorithm will likely use them as cover images.

3.2. The steganographic algorithm

There are several steganographic algorithms working directly in the JPEG domain. In our research we used the MB1 algorithm proposed in [6]. While there is no particular reason behind this choice, the general conclusions of our paper remaining the same for other steganographic methods, we found that the MB1 algorithm is particularly suited to highlight the merits and drawbacks of FA-steganalysis. The MB1 algorithm has been used to process all the images of \( D_1 \) and \( D_2 \) datasets, embedding a message with a payload of 0.05 bpc (bits per non-zero DCT coefficients).

3.3. The steganalyzer

Among the steganalyzers designed to work with JPEG images, we selected the blind steganalyzer proposed by Pevny et al. [7] for the good performance it ensures across a wide variety of steganographic algorithms. In their work, Pevny et al. summarize the current state of art steganalyzers by merging features extracted from the DCT domain [8] and by using Markov analysis on neighborhood of DCT coefficients [9]. The first set of features [8] is composed by 193 features and it is built around a deep analysis of the statistics of DCT coefficients. The second set of features models the differences between absolute values of neighboring DCT coefficients as a Markov process like in [9]. In addition, all 274 features undergo a calibration phase consisting in taking the analyzed JPEG image decompress and crop it by 4 pixels in each direction. Then the image is compressed again with the same quantization table as the initial one. Cropping and recompression should produce a calibrated image with macroscopic features similar to the original cover image. As opposed to Pevny’s steganalyzer we used a classifier based on Fisher linear discriminant instead of an SVM.

3.4. Telling CG and camera images apart

As a final step we must select a classifier able to distinguish between CG and camera images. This is a rather well studied problem in image forensics for which several solutions exist. Among them we chose the algorithm proposed by Ng et al. in [10]. The main characteristic of such a scheme is that of relying on the geometric features of the to-be-analyzed images thus reducing the dependence of the classification result on subtle statistical features like noise statistics.

4. EXPERIMENTAL RESULTS

In this section we report the experimental results that we obtained by applying the FA-steganalyzer (\( S_{FA} \)) built as described above to \( D_1 \) and \( D_2 \), and compare them against those obtained by a mixed steganalyzer (\( S_{mix} \)). Throughout all the experiments, the datasets have been split into a training and a test set, the former containing 50% of the images. In addition, cross-validation of the results has been carried out, i.e. the experiments have been carried out 10 times by randomly choosing the training and test dataset. The error probabilities given below, hence, are average error probabilities, while for ROC curves vertical averaging has been used and the variability of \( P_N \) for a given \( P_F \) given.

4.1. Homogeneous vs mixed training

We start by considering the performance of the steganalyzers when they are trained on homogeneous datasets, i.e. according to the symbolism used in section 2.1, we analyze the performance of \( S_1 \) and \( S_2 \). A thorough discussion of such performance requires that the ROC curves of the two steganalyzers are given. For sake of brevity, however, we report only the results corresponding to the AUC (Area Under the Curve), since it provides a good summary of the goodness of the whole ROC curve. Specifically, Table 1 reports the AUC obtained on \( D_1 \) and \( D_2 \). The datasets \( D_1 \) and \( D_2 \) have been used consistently, i.e. steganalyzers trained on \( D_1 \) (res. \( D_2 \)) have been tested on \( D_1 \) (res. \( D_2 \)). As expected, the error rate tends to increase when a steganalyzer is used to analyze images belonging to an image class that is different from that used during the training phase.

Table 1 gives also the results obtained by \( S_{mix} \). For the \( D_1 \) dataset, training the steganalyzer on a mixed dataset does not help very much, while for \( D_2 \) the performance of the mixed steganalyzer are very close to the results obtained when each steganalyzer works on images belonging to the dataset on which it was trained.

4.2. Forensics-aided steganalysis

In order to assess the performance of the FA-steganalyzer the error probabilities of the image forensics classifier must be evaluated. We did so for both datasets obtaining the results reported in figure 3. Even in this case the datasets \( D_1 \) and \( D_2 \) have been used consistently. As expected, the performance of the classifier are much better when dataset \( D_1 \) is used. In order to use the analysis given in section 2.1 to evaluate the performance of the FA-steganalyzer, it is necessary that an operating point is chosen for its three components, namely \( S_1 \), \( S_2 \) and \( C \). This is equivalent to setting the decision threshold of the corresponding classifiers. The various error probabilities should then be used within the equations derived in section 2.1 to obtain the overall error probability. As to \( S_{mix} \) the corresponding error probability is simply the average of the error probabilities obtained on \( C_1 \) and \( C_2 \).
4.3. ROC curves

While the procedure outlined above is a very simple one, drawing the ROC curves of the FA-steganalyzer is not straightforward, since its operating point is determined by the three decision threshold of its components. Indeed for each value of the false detection probability one should determine the optimum value of the three thresholds so that the correct detection probability is maximized. This is not an easy task due to the lack of a simple expression giving the dependency of the various error probabilities appearing in equation (2) on the decision thresholds of $S_1$, $S_2$ and $C$. For this reason, we adopted an experimental procedure working as follows. We fixed a minimum and maximum value for each decision threshold and then evaluated the performance of the steganalyzer for all the combinations of the thresholds within the considered interval, quantized with at a suitable step. The results is a cloud of points in the $P_D$ vs $P_F$ diagram as reported in figure 4. For each value of the false detection probability, we considered the point resulting in the higher correct detection rate, since this is the optimal parameter configuration for the given $P_D$. The line connecting all the points found in this way (see Fig. 4) gives the optimal ROC curve for the steganalyzer. Such a curve can now be compared with the ROC curve of $S_{mix}$ given in figure 5, demonstrating the superior performance obtained by $S_{FA}$.

Interestingly, $S_{FA}$ outperforms $S_{mix}$ even for $D_2$. This may seem a surprising result, since in this case the performance of $S_{mix}$ are very similar to those that would be obtained by letting each image be analyzed by the correct steganalyzer (see table 1), i.e. the performance of an FA-steganalyzer equipped with an error-free forensics classifier $C$. The explanation for such a surprisingly good behavior is that an error of the forensics classifier does not necessarily result in an error of the steganalyzer. Indeed, though on the average images in $C_1$ (res. $C_2$) are better steganalyzed by $S_1$ (res. $S_2$) this may not be the case for some particular images, thus making an error of the forensics classifier desirable. Indeed this is likely to be the case for images in $C_1$ (res. $C_2$) whose statistics are more similar to the images in $C_2$ (res. $C_1$), i.e. exactly for the images for which the forensics operator is more likely to make an error.

![ROC curves of the FA-steganalyzer applied to $D_1$ (left) and $D_2$ (right).](image)

**5. CONCLUSIONS**

In this paper we considered the use of an image forensics tool for easing the steganalysis of images produced by different sources. By adapting the steganalyzer used to the class the image at hand belongs to, it is, in fact, possible to significantly increase the classification performance. Though we have demonstrated the validity of the considered approach in a rather simple set up involving only two image classes for which good forensics instruments exist, we believe that FA-steganalysis may be helpful in a wide variety of cases. Further works should compare more accurately FA-steganalysis and mixed-training steganalysis; while it is clear that in the presence of accurate forensics tools FA-steganalysis is preferable, the exact limits of this approach are not clear yet. The development of FA-steganalyzers capable of working in scenarios more complex than the one considered in this paper, is another direction for future research.

**6. REFERENCES**


