Breaking HUGO – the process discovery

Jessica Fridrich, Jan Kodovský, Vojtěch Holub, and Miroslav Goljan

Department of ECE, SUNY Binghamton, NY, USA
{fridrich,jan.kodovsky,vholub1,mgoljan}@binghamton.edu

Abstract. This paper describes our experience with the BOSS competition in chronological order. The intention is to reveal all details of our effort focused on breaking HUGO – one of the most advanced steganographic systems ever published. We believe that researchers working in steganalysis of digital media and related fields will find it interesting, inspiring, and perhaps even entertaining to read about the details of our journey, including the dead ends, false hopes, surprises, obstacles, and lessons learned. This information is usually not found in technical papers that only show the final polished approach. This work accompanies our other paper in this volume [9].

1 Introduction

Competitions, such as BOSS (Break Our Steganographic System) [5] or BOWS (Break Our Watermarking System) [2] help focus the attention of the research community to a specific problem and thus advance the field by a large margin within a rather short time span. This is because challenges and competitive environment have always appealed to humans and also due to the fact that the participants do not need to formulate the problem (a task that is sometimes more important than the solution). Moreover, the competition guarantees that the results of different teams are comparable. For BOSS, the performance is evaluated using a single scalar value – the BOSSrank score.

According to our understanding, the original intention behind BOSS was to investigate whether content-adaptive steganography improves steganographic security for empirical covers in the form of raster, never-compressed images. The fact that in adaptive steganography the selection channel (placement of embedding changes) is publicly known, albeit in a probabilistic form, could in theory be exploited by an attacker. Adaptive schemes also introduce more embedding changes than non-adaptive schemes because some pixels are almost forbidden from being modified. Thus, an adaptive scheme will embed with a larger change rate than a non-adaptive one. On the other hand, the changes are constrained to those regions of images that are hard to model and thus the change rate is not an appropriate measure of statistical detectability as it puts the same weight to all pixels. The organizers of BOSS proposed a different distortion measure and argued that it better corresponds to detectability of embedding. To further substantiate their claim, the measure was incorporated in the steganographic algorithm HUGO (Highly Undetectable SteGO) [16] and the stego community was challenged to attack it. Preliminary tests with existing steganalyzers indeed indicated that HUGO is significantly more resistant to steganalysis than previous algorithms.

The BOSS competition, including the rules and the materials made available to the competitors, is described in a different paper in this volume [1]. Our team entered the competition at the end of August. This paper reveals the details of our investigation in chronological order. This technical narrative will hopefully be inspiring and maybe even amusing to those who tried to break HUGO and, in general, to all interested in steganalysis of digital media. Portraying our effort including the final results as well as our false beliefs and dead ends will convey those aspects of research work that is typically not found in technical papers. Our understanding of the field has evolved much over the last few months. We were forced to abandon established paradigms and reevaluate existing empirical truths. As a result, we learned quite a bit and we certainly hope that the reader of this paper will as well. This paper accompanies another paper [9] in this volume, which contains additional technical details of our final approach together with an extensive experimental section.

Everywhere in this article, lower-case boldface symbols are used for vectors and capital-case boldface symbols for matrices or higher-dimensional arrays. The symbols $\mathbf{X} = (x_{ij}) \in \mathcal{X} = \{0, \ldots, 255\}^{n_1 \times n_2}$ and $\mathbf{Y} = (y_{ij}) \in \mathcal{X}$ will always represent pixel values of grayscale cover and stego images with $n = n_1 n_2$ pixels. When the two-dimensional character of the pixel arrays is not important, for convenience, and hopefully without introducing any confusion, we index pixels with a single symbol instead of a pair. We will use $E[X]$ and $Var[X]$ for the expected value and variance of random variable X. For any $x \in \mathbb{R}$, the largest integer smaller than or equal to x is $\mathrm{floor}(x)$. The detection accuracy of steganalyzers will always be evaluated on a test set never seen by the steganalyzer using a scalar score defined as

$$\rho \triangleq 1 - \min_{P_{\mathrm{FA}}} \frac{1}{2}(P_{\mathrm{FA}} + P_{\mathrm{MD}}(P_{\mathrm{FA}})), \tag{1}$$

where P_{FA} and P_{MD} are the probabilities of false alarm and missed detection. When the score is computed from BOSSrank images, it will always be referred to as the "BOSSrank score."

2 Early ideas – is the public selection channel a problem?

The very first idea that naturally lends itself is whether it is possible to somehow utilize the fact that the attacker can approximately determine the probabilities with which each pixel was changed during embedding. According to the folklore, revealing where embedding changes are made and where they are not may be a weakness of adaptive embedding that may be exploited.

HUGO modifies pixel x_i by ±1 with probability $p_i^{\mathbf{X}}$ that can be determined from the cover image $\mathbf{X}$ and the payload.[1] Since the source code of HUGO

[1] For details, see [1] in this volume.

is public, one can easily extract the algorithm that computes the probabilities. However, when the image inspected by the attacker is a stego image, the probabilities computed from the stego image will in general be slightly different $p_i^{\mathbf{Y}} \neq p_i^{\mathbf{X}}$. Fig. 1 shows $p_i^{\mathbf{Y}}$ versus $p_i^{\mathbf{X}}$, $i = 1, \ldots, n$, for BOSSbase image no. 50. Overall, $p_i^{\mathbf{X}} \approx p_i^{\mathbf{Y}}$ with the largest relative errors for small $p_i^{\mathbf{X}}$. In particular, $|p_i^{\mathbf{Y}} - p_i^{\mathbf{X}}| \leq 0.05$ for 99.4% of pixels, $|p_i^{\mathbf{Y}} - p_i^{\mathbf{X}}| \leq 0.01$ for 85.2% pixels, and $|p_i^{\mathbf{Y}} - p_i^{\mathbf{X}}| \leq 0.001$ for 40.4% of pixels.

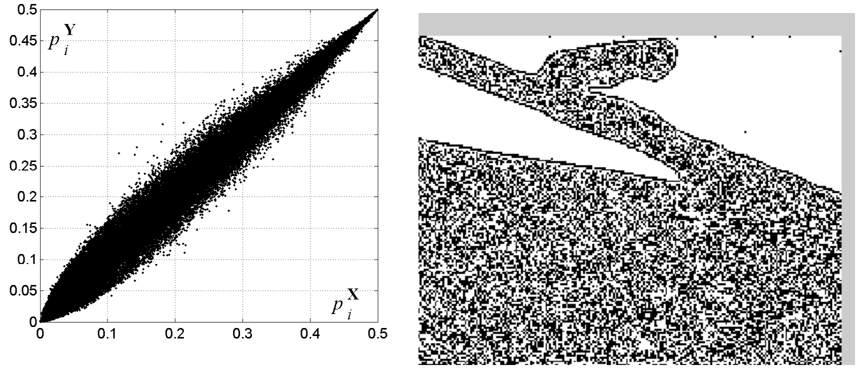


Fig. 1. Left: Probability of embedding change computed from the stego image, $p_i^{\mathbf{Y}}$, vs. $p_i^{\mathbf{X}}$ (for image no. 50 from BOSSbase). Right: LSB plane of the upper-right corner of image no. 235 from BOSSrank. The embedding changes are visible as black dots around the image boundary.

Because the payload is known and because $p_i^{\mathbf{X}} \approx p_i^{\mathbf{Y}}$, one could in theory derive (at least in expectation) the values of cover-image statistics, such as histograms or co-occurrence matrices. However, even if we succeeded in accurately estimating the cover-image statistics, using these estimates for steganalysis may still be problematic because HUGO does not introduce any easily detectable changes and we may not have any way of telling whether we are inspecting a cover or a stego image.

Having abandoned this direction, it is rather amusing that HUGO's embedding changes can be detected *visually* in seven images from BOSSrank – images no. 62, 195, 235, 396, 438, 948, and 983.[2] All seven images contain a region of pixels saturated either at 255 or at 0 while the rest of the image lacks any complex texture. Since HUGO was forced to embed 0.4 bpp in every image and since the probability of embedding in saturated areas is not completely zero, the embedding leaves suspicious salt-and-pepper noise in the least significant bit plane. An example is shown in Fig. 1 right. Notice that most of the visible

[2] These images were all classified correctly using our feature-based approach described below in this paper, thus the visual attack did not help us increase our BOSSrank score.

embedding changes are concentrated around the image boundary – most likely a consequence of how the embedding probabilities are computed at boundary pixels.

2.1 Detection by correlation?

If we were able to estimate from the stego image whether a given pixel was modified by 1 or -1 with probability better than random guessing, we could detect HUGO (and ±1) embedding using a correlation just like a spread-spectrum watermark. This idea is essentially identical to the Weighted Stego steganalysis [8]. Using $y_i = x_i + s_i$, $s_i \in \{-1, 0, 1\}$, we have $1/n \sum_i s_i^2 = \beta$, the change rate. (For HUGO with payload 0.4 bpp, $\beta \approx 0.1$, depending on the content.) Furthermore, let $\hat{x}_i = x_i + \varXi_i$ be an estimate of x_i from $\mathbf{Y}$ (e.g., $\hat{x}_{ij} = (y_{i,j-1} + y_{i,j+1})/2$), with $\varXi_i$ being the estimation error. Assuming that the embedding change s_i can be estimated from $\mathbf{Y}$ with probability better than random guessing, i.e., $\sum_{i=1}^{n} \hat{s}_i s_i \propto bn\beta$ with $b > 0$, we now analyze the following correlation for a cover and a stego image:

$$\rho = \sum_i (y_i - x_i)\hat{s}_i = \sum_i (s_i - \varXi_i)\hat{s}_i = \sum_i s_i\hat{s}_i - \sum_i \varXi_i\hat{s}_i. \qquad (2)$$

When $\mathbf{Y}$ is a stego image and if $\boldsymbol{\varXi}$ and $\hat{\mathbf{s}}$ are uncorrelated, $E[\rho] \propto n$ while for a cover image $\mathbf{X}$, $E[\rho] \approx 0$. Also, $Var[\rho] \propto n$ in both cases. This opens the possibility to detect embedding by thresholding ρ. This idea, however, hinges upon two assumptions – that we can estimate the direction of an embedding change with probability better than random guessing and the assumption of $\boldsymbol{\varXi}$ and $\hat{\mathbf{s}}$ being uncorrelated. While it is, indeed, possible to estimate $\hat{s}_i$ with probability better than random guessing, for example by testing if a change of y_{ij} by 1 or -1 decreases the sum $\sum_{0<|a|+|b|\le 2} |y_{ij} - y_{i+a,j+b}|$, $\boldsymbol{\varXi}$ and $\hat{\mathbf{s}}$ are, unfortunately, correlated. The reason is the content-adaptive character of embedding. As a result, even though $E[\rho(\mathbf{Y})] > E[\rho(\mathbf{X})]$, it is not possible to find a threshold for ρ as it varies greatly across images. For some images, we observed the increase in correlation up to 60% but the average increase (over BOSSbase) was only 1.74%, which is by several orders of magnitude smaller than the variations of ρ across images.

3 Pixel domain is not useful, right?

HUGO preserves complex statistics in a 10^7-dimensional feature space built from joint statistics of pixel differences on 7×7 neighborhoods. Thus, it may seem that features computed from differences between neighboring pixels will lead to weak detection simply because the embedding algorithm was designed to preserve statistics in this domain. One argument supporting this point of view is the experimental result reported in the original publication [16]: While the performance of the second-order SPAM feature set [15] (dimensionality 686) on HUGO is quite weak ($\rho = 58\%$), after augmenting SPAM with the DCT-based

Cartesian-calibrated Pevný set [13] (dimensionality 548), the score improved to $\rho = 65\%$.[3] This line of reasoning initially motivated us to compute features in an alternative domain, such as the wavelet domain. To this end, we decided to modify the WAM feature vector originally introduced in [10].

The WAM features are computed by first transforming the image to the wavelet domain using the Daubechis D8 wavelet, $(\mathbf{H}, \mathbf{V}, \mathbf{D}, \mathbf{L}) = W(\mathbf{X})$. When an undecimated transform is used, the first-level wavelet transform produces four subbands, $\mathbf{H}, \mathbf{V}, \mathbf{D}, \mathbf{L}$, of the same size as the original image. The three high-frequency subbands, $\mathbf{H} = (h_{ij}), \mathbf{V}, \mathbf{D}$, are denoised using the Wiener filter with variance σ_{W}^2:

$$\hat{h}_i = \frac{\hat{\sigma}_i^2}{\hat{\sigma}_i^2 + \sigma_{\mathrm{W}}^2} h_i, \tag{3}$$

where $\hat{\sigma}_i^2$ is the local variance at wavelet coefficient i estimated from its neighborhood. Finally, the WAM features, $\mu_m^{\mathrm{h}}, \mu_m^{\mathrm{v}}, \mu_m^{\mathrm{m}} \in \mathbb{R}^9$, are formed as nine central moments of their corresponding high-frequency subband noise residuals:

$$\mu_m^{\mathrm{h}} = \frac{1}{n} \sum_{i=1}^{n} \left| \hat{h}_i - h_i - (\overline{\hat{\mathbf{H}} - \mathbf{H}}) \right|^m, \quad m \in \{1, \ldots, 9\}, \tag{4}$$

which gives a feature vector of dimension 27.

Our initial tests were done on BOSSbase 0.9 containing $2 \times 7,518$ images. The database was randomly divided into two equal-size subsets, one used for training and the other for testing. A Gaussian Support Vector Machine (G-SVM) was trained using standard five-fold crossvalidation on a multiplicative grid. The original WAM classifier with default $\sigma_{\mathrm{w}}^2 = 1/2$ gave the score of $\rho = 55.85\%$. To improve this rather weak performance, we decided to extend WAM by adding 27 moments (4) computed directly from the subbands $\mathbf{H}, \mathbf{V}, \mathbf{D}$ to inform the steganalyzer about the image content. This, indeed, makes sense to do for spatially-adaptive steganography. This content-informed WAM feature (WAMC) of dimensionality 54 reached the score of $\rho = 57.40\%$.

Exploring a different extension of WAM features, we augmented them with the same feature computed from an image re-embedded with the same payload of 0.4 bpp. This 54-dimensional vector (WAMre) produced a respectable $\rho = 59\%$.

The final and most significant improvement of WAM involved replacing the Wiener filter (3) with its adaptive version in which the fixed noise variance σ_{w}^2 was replaced with the variance of the stego noise s_i at pixel i, $\sigma_{\mathrm{w},i}^2 = p_i^{\mathbf{Y}}$. The best performance was achieved by merging the original 27 WAM features, 27 content features, and 27 WAM features obtained using the adaptive filter and adding to it the same set of 81 features from a re-embedded image (total of 162 features WAMCPre). The final performance is summarized in Table 1.

As part of our investigation of alternative embedding domains, we also tested the Cross-Domain Features (CDF) [13], which is a merger of the second-order SPAM with Cartesian-calibrated Pevný set (total dimensionality 1234). A G-SVM produced a prediction file with BOSSrank score of 68%, which is higher

[3] This result was reported on the BOWS2 database [2].

Feature	Dimension	Score ρ [%]
WAM	27	55.85
WAMC	54	57.40
WAMre	54	59.00
WAMCPre	162	62.97

Table 1. Performance of the WAM steganalyzer and its various extensions.

than the score of 65% obtained using the same feature set reported by the Czech University Team. This difference is most likely caused by a different training set. While we trained on all images from BOSSbase 0.91, the Czech University Team trained on one half of this database.

4 Going back to pixel domain

Even though alternative domains may be useful in steganalysis, the best detection is usually achieved by forming features *directly in the embedding domain*. This is where the embedding changes are localized and thus most pronounced. This strategy, originally coined in 2004 [6], was later confirmed in [6, 10, 17, 15, 18, 4]. Because HUGO's embedding domain is known, after the early failures described in the previous two sections, we revisited the pixel domain and achieved a major breakthrough on September 23, 2010.

HUGO approximately preserves the joint distribution of first-order differences $r^{(1)}_{ij} = x_{i,j+1} - x_{ij}$ between four neighboring pixels – the co-occurrence of triples $(r^{(1)}_{ij}, r^{(1)}_{i,j+1}, r^{(1)}_{i,j+2})$ truncated[4] to a finite dynamic range, $r_{ij} \leftarrow \mathrm{trunc}_T(r_{ij})$, where $\mathrm{trunc}_T(x) = x$ when $x \in [-T, T]$ and $\mathrm{trunc}_T(x) = T\mathrm{sign}(x)$ otherwise. Thus, to detect traces of embedding, a fourth-order co-occurrence $(r^{(1)}_{ij}, r^{(1)}_{i,j+1}, r^{(1)}_{i,j+2}, r^{(1)}_{i,j+3})$ is needed. However, with increasing order of the co-occurrence its elements will be rather sparse when computed from small images and thus too noisy for steganalysis. The *key* idea and a major breakthrough in our effort to break HUGO was the realization that another way to form a statistic that spans more than four pixels is to use *higher-order pixel differences (residuals)*.

Because the second-order residuals, $r^{(2)}_{ij} = x_{i,j-1} - 2x_{ij} + x_{i,j+1}$, involve three pixels, one needs to consider the joint statistic of only three adjacent differences $(r^{(2)}_{ij}, r^{(2)}_{i,j+1}, r^{(2)}_{i,j+2})$. This keeps the co-occurrence matrix well-populated and thus useful for detection. The second-order residuals better remove content that is locally linear – while $r^{(1)}_{ij}$ may get out of the dynamic range $[-T, T]$ in locally linear regions, $r^{(2)}_{ij}$ may be mapped back inside the interval $[-T, T]$. One can also interpret $r^{(2)}_{ij} = 2(\hat{x}_{ij} - x_{ij})$, where $\hat{x}_{ij} - x_{ij}$ is the noise residual at pixel ij obtained using a simple denoising filter that predicts the value of the central pixel as an arithmetic average of its two closest neighbors: $\hat{x}_{ij} = \frac{1}{2}(x_{i,j-1} + x_{i,j+1})$. It

[4] The truncation is an established way to keep the dimensionality low prior to forming joint statistics.

is very important that the denoised value does not depend on the central pixel in any way, otherwise $\hat{x}_{ij}$ would be affected by the stego signal s_{ij}, which would thus be undesirably suppressed in $r_{ij}^{(2)}$.

Before describing the first successful feature set that gave us BOSSrank over 70%, we introduce four types of operators that can be applied to any two-dimensional array $\mathbf{A} = (a_{ij})$. The horizontal co-occurrence is a matrix $C^{\mathrm{h}}(\mathbf{A})$ whose (d_1, d_2, d_3)th element, $d_1, d_2, d_3 \in [-T, T]$, is

$$C^{\mathrm{h}}_{d_1 d_2 d_3}(\mathbf{A}) = |\{(i,j)|(a_{i,j}, a_{i,j+1}, a_{i,j+2}) = (d_1, d_2, d_3)\}|. \tag{5}$$

The operators C^{v}, C^{d}, and C^{m} are defined analogically.

After many initial experiments, we arrived at the following two feature vectors that allowed us to improve our BOSSrank score by a rather large margin. First, compute four second-order residuals at each pixel along the horizontal, vertical, diagonal, and minor diagonal direction:

$$\begin{aligned} r^{\mathrm{h}}_{ij} &= x_{i,j-1} - 2x_{ij} + x_{i,j+1}, & r^{\mathrm{v}}_{ij} &= x_{i-1,j} - 2x_{ij} + x_{i+1,j}, \\ r^{\mathrm{d}}_{ij} &= x_{i-1,j-1} - 2x_{ij} + x_{i+1,j+1}, & r^{\mathrm{m}}_{ij} &= x_{i-1,j+1} - 2x_{ij} + x_{i+1,j-1}. \end{aligned} \tag{6}$$

and then form the MIN and MAX residuals:

$$r^{\mathrm{MIN}}_{ij} = \mathrm{trunc}_T(\min\{r^{\mathrm{h}}_{ij}, r^{\mathrm{v}}_{ij}, r^{\mathrm{d}}_{ij}, r^{\mathrm{m}}_{ij}\}) \qquad r^{\mathrm{MAX}}_{ij} = \mathrm{trunc}_T(\max\{r^{\mathrm{h}}_{ij}, r^{\mathrm{v}}_{ij}, r^{\mathrm{d}}_{ij}, r^{\mathrm{m}}_{ij}\}). \tag{7}$$

The MINMAX feature vector is defined as

$$\mathbf{F}^{\mathrm{MINMAX}} = (C^{\mathrm{h}}(\mathbf{R}^{\mathrm{MIN}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{MIN}}), C^{\mathrm{h}}(\mathbf{R}^{\mathrm{MAX}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{MAX}})). \tag{8}$$

Since each cooccurrence matrix has $(2T+1)^3$ elements, $\mathbf{F}^{\mathrm{MINMAX}}$ has dimensionality of $2(2T+1)^3$.

By training the MINMAX feature vector with $T = 4$ using Fisher Linear Discriminant (FLD) on 9,074 cover and stego images from BOSSbase 0.91, we achieved a BOSSrank score of 71% on October 3, 2010.

The next discovery we made can be interpreted as a clever marginalization of the MINMAX vector for $T = 8$. Before forming r^{MIN}_{ij} and r^{MAX}_{ij}, the differences are quantized using a scalar quantizer $Q_q(x) = \mathrm{floor}(x/q)$ with q a positive integer:

$$\begin{aligned} &\mathbf{F}^{\mathrm{QUANT},q} = \\ &\left(C^{\mathrm{h}}(Q_q(\mathbf{R}^{\mathrm{MIN}})) + C^{\mathrm{v}}(Q_q(\mathbf{R}^{\mathrm{MIN}})), C^{\mathrm{h}}(Q_q(\mathbf{R}^{\mathrm{MAX}})) + C^{\mathrm{v}}(Q_q(\mathbf{R}^{\mathrm{MAX}}))\right). \end{aligned} \tag{9}$$

For $q = 2$, this QUANT feature can “see” twice as far as MINMAX but in a quantized manner to keep the dimensionality of the feature unchanged. By training a G-SVM on BOSSbase 0.91 on the 2,916-dimensional feature vector $(\mathbf{F}^{\mathrm{MINMAX}}, \mathbf{F}^{\mathrm{QUANT},2})$, with $T = 4$, we obtained a BOSSrank score of 73% on October 4, 2010.

On October 11, the organizers announced that the first 7,518 stego images from BOSSbase 0.9 and 0.91 were created with a different set of parameters

($\sigma = 10$, $\gamma = 4$, see [16] or [1] for details of the embedding algorithm) than all BOSSrank stego images and the rest of the stego images in BOSSbase 0.91 (which were created with $\sigma = 1$, $\gamma = 1$). This change in parameters caused a mismatch between the training and testing stego sources. After recomputing the MINMAX and QUANT features on the correct stego images, on October 12 we achieved the score of 75% by merging the MINMAX and QUANT into a 2,916-dimensional feature set. Thus, the drop of performance due to this stego-source mismatch was 2%. To us, it was a HUGE difference even though the BOSS Team claimed on their blog on October 11 that HUGO behaves "similarly" for both choices of the parameters.

At this point, our team became confident that the 80% milestone was within reach by the end of October. We could not have been more wrong! Not only have we become hopelessly stuck at 75% for more than a month, but it would take us two and half months of very hard work to reach 80%. And we did so on December 23 with a feature vector of dimensionality $22,307$ trained on $2\times 24,184$ images. To be able to train a classifier at this scale, we had to abandon SVMs and reinvent the entire machine learning approach. But before we get to that, in the next section we describe the Warden's nightmare.

5 The dreaded cover-source mismatch

The next logical step in our attack was to fine-tune our feature set by finding the optimal value of the threshold T, adding other versions of the features, and perhaps by training on a larger number of images. We also moved to a four-dimensional co-occurrence operator for the QUANT feature set, obtaining thus a 4,802-dimensional feature vector ($2 \times (2 \times 3 + 1)^4 = 4802$). To our big surprise, while we steadily improved detection accuracy on BOSSbase by adding more features, the BOSSrank score was moving in the *opposite* direction. We began facing the dreaded cover-source mismatch issue[5] – our classifier was trained on a different source of cover images (BOSSbase) than the source of BOSSrank images. Thus, as we optimized our detector on the training set, the performance on the testing set was steadily worsening. Our detector lacked what is recognized in detection theory as robustness.

Google search on "robust machine learning" returned publications that concerned only the case of training on noisy data or on data containing outliers. Our problem seemed different – we trained on one distribution and tested on another.

Perhaps using classifiers with less complicated decision boundary than the one produced by a G-SVM might help. The performance of a linear SVM (L-SVM), however, was consistently subpar to G-SVM and disturbingly comparable to the much simpler FLD classifier (see Table 2).

Another way to increase robustness, or so we thought, was to train on a larger set of images. We added to BOSSbase 0.91 another set of 6,500 images taken

[5] Cover source mismatch differs from overtraining as the latter refers to the lack of ability of the detector to generalize to unseen examples from the same source.

Feature	Dimensionality	Training set	G-SVM	L-SVM	FLD
MINMAX	1458	BOSSbase 0.92	73	70	71
QUANT	1458	BOSSbase 0.92	73	72	71
MINMAX+QUANT	2916	BOSSbase 0.92	75	72	71
MINMAX+QUANT	2916	BOSSbase+CAMERAS	71	70	-

Table 2. BOSSrank score of the first successful feature sets, MINMAX and QUANT, for three different machine learning approaches.

in raw format by 22 different cameras converted using the same script that was used for creating the BOSSbase. Training on more images, however, seemed to make the BOSSrank score only worse (see the last row in Table 2).

The cover-source mismatch has been recognized by the research community as a serious issue that may complicate deployment of steganalysis in real life. The authors of [10] reported that the performance of the WAM steganalyzer on images could be vastly improved if the steganalyzer was trained on images from the exact same camera or, to a slightly lesser degree, on images from a camera of the same model. However, training WAM on a mixture of images from CAMERAS, the performance was significantly worse. The cover-source mismatch problem was also mentioned in the more recent publication [3], where the authors tested various steganalyzers on multiple sources for the ±1 embedding. Thus, as the next logical step in our quest we decided to find out as much as possible about the source of covers for BOSSrank. We saw this as the only way to further improve our BOSSrank score.

5.1 Forensic analysis of BOSSrank

On October 14, we extracted the sensor fingerprint [7] for each camera from BOSSbase (we did so from the resized grayscale 512×512 images). Then, we tested all BOSSrank images for the presence of the fingerprints. Only one camera tested positive – the Leica M9. Its fingerprint was found in approximately 490 images. We knew the source of one half of the database.

Visual inspection of BOSSrank images revealed that at least some portion of images was taken in the Pacific Northwest because many pictures contained license plates from the State of Oregon and Washington. One image (see Fig. 2 upper left) contained an address, which, after plugging it in GoogleMaps, returned the exact location – Portland, Oregon. And after the photographer was identified in a window pane reflection in image no. 558 (see Fig. 2 right), we knew what the camera was – Panasonic Lumix DMC-FZ50 – and it belonged to Tomáš Filler, a BOSS Team member.[6] However, we could not use this finding in competition because we relied on information other competitors did not have access to. Therefore, we closed our forensic investigation knowing that roughly one half (and potentially more) BOSSrank images were from Leica M9. The source

[6] The camera was confirmed by identifying its fingerprint in about 90 BOSSrank images. Here, we extracted the fingerprint from other images taken by Tomáš Filler during our previous trips to the SPIE conference.

of the remaining images in BOSSrank was declared unknown. All we needed to do now was to obtain more images from Leica.

Since stealing the camera from Patrick Bas seemed too dangerous and buying it too expensive ($7,000), we rented it from http://www.lensrentals.com/ for a week (October 23–30). The camera was rented with the standard 50mm lens.[7]After a grueling work with a heavy and boxy camera with no auto focus, we managed to take a total of 7,301 images in their original resolution of 18 megapixels. All images were processed using the BOSS conversion script and subsequently embedded with payload 0.4 bpp. After the MINMAX+QUANT features were extracted from them, we built two detectors – one G-SVM trained on all BOSSbase images that would be used for detection of all non-Leica images from BOSSrank, and the second G-SVM specifically trained on the union of our 7,301 Leica images and the 2,267 Leica images from BOSSbase. The decisions would then be merged into one prediction file. The result was quite disheartening – a measly 74% (BOSSrank score). We ran a couple of more experiments, such as training a G-SVM on a union of BOSSbase and 7,301 Leica images and testing the entire BOSSrank with it, but none of these experiments would produce a BOSSrank score higher than 74%.

This rather time-consuming exercise was an important lesson for us because we realized what makes a cover source and how hard it is to duplicate it. First, we took images with a different lens (50mm) than the BOSSbase images (35mm). The lens may have a significant impact on steganalysis because a longer focal length means lower depth of field, which implies less content in focus and more content slightly out of focus. Of course, an out-of-focus content is easier for the steganalyst.

The content of images has obviously a major influence on content-adaptive steganalysis. The cover source is a very complex entity that is affected by the lens, the environment in which pictures are taken and even the photographer's habits – stopping the lens more leads to a higher depth of field but also darker images with potentially more motion blur, while opening the lens leads to shorter exposures and less dark current but lower depth of field. Our images were all taken in the Fall in a little town of Binghamton in upstate New York. On the contrary, a large number of the Leica images in BOSSrank showed scenes with an ocean, ships, beaches, etc. As one of us sighed: "Binghamton in the Fall is a poor replacement for French Riviera." Consequently, it was rather foolish to think that we could duplicate the cover source.

6 Diversity is important

One important lesson we learned by now is that one should not be afraid of high feature dimensionality. After all, we successfully trained a 2,916-dimensional feature vector on $2\times 9,074$ images and obtained a high BOSSrank score. However, scaling up the dimensionality simply by increasing the threshold T or the order of

[7] Only later it was pointed out to us that the lens information is in the EXIF headers of BOSSbase images. And the lens used for BOSSbase had a focal length of 35mm.

Fig. 2. Identifying the source of BOSSrank.

the co-occurrence matrix did not lead to better results because the added features were increasingly sparsely populated. Thus, we refocused our effort to creating a more *diverse* feature set while keeping the dimensionality around 3,000, what seemed as a sweet spot for the given training set (BOSSbase). To this end, we used a lower threshold $T = 3$ and incorporated higher-order differences among neighboring pixels. One can easily extend the MINMAX and QUANT feature vectors (8) and (9) to higher-order residuals:

$$r_{ij}^{(3)} = x_{i,j-1} - 3x_{ij} + 3x_{i,j+1} - x_{i,j+2} \tag{10}$$

$$r_{ij}^{(4)} = -x_{i,j-2} + 4x_{i,j-1} - 6x_{ij} + 4x_{i,j+1} - x_{i,j+2}. \tag{11}$$

We also built features using fourth-order co-occurrence operators. To limit the growth of feature dimensionality, we used $T = 2$ for all fourth-order co-occurrences. This reasoning gave birth to the following 3,872-dimensional feature set SUM3 consisting of four different subsets (see Table 3).

Difference order	q	Cooc. order	T	Dimensionality
2nd	2	3	3	686
3rd	2	3	3	686
2nd	2	4	2	1,250
3rd	2	4	2	1,250

Table 3. A merger of four feature sets, SUM3, computed from second- and third-order differences among pixels forming co-occurrence matrices of order 3 and 4. The feature dimensionality is 3,872.

The strategy of increasing feature diversity was successful. By training a G-SVM on images from BOSSbase and with the feature set shown in Table 3 we obtained a BOSSrank score of 76% on November 13. The direction that was opening for us was clear – instead of blindly increasing the threshold and co-occurrence order, increase the feature diversity! For example, one could form higher-order residuals (differences) using two-dimensional kernels instead of one-dimensional or extract the residuals along edges to improve detection for textured images. The complexity of training a G-SVM, however, was beginning to limit the speed of development, while the performance of the much faster L-SVMs was subpar compared to G-SVMs. We needed an alternative machine learning tool that would enable faster development and testing of many ideas and combinations of features. Fortunately, our other research direction that we were simultaneously pursuing independently of the BOSS competition gave us just what we needed – an inexpensive, fast, and scalable machine learning approach.

7 Ensemble classifiers – a great alternative to SVMs

In this section, we only provide a rather brief description, referring to [14] and our other paper in this volume [9] for a more detailed exposition of this methodology, experimental evaluation and comparison to SVMs as well as a discussion on the relationship of our approach to prior art in machine learning.

Starting with a feature set of full dimensionality d, we build a simple classifier (base learner), such as an FLD, on a randomly selected subset of $d_{\text{red}} \ll d$ features while using all training images. The classifier is a mapping $F : \mathbb{R}^d \to \{0, 1\}$, where 0 and 1 stand for cover and stego classes. This is repeated L times with a different random subset of the features. Consequently, we obtain L classifiers (FLDs) $F_1, \ldots, F_L$. Given a feature vector $\mathbf{b} \in \mathbb{R}^d$ from the testing set, the ensemble classifier makes a decision by fusing the individual decisions of all L FLDs, $F_1(\mathbf{b}), \ldots, F_L(\mathbf{b})$. Although many fusion rules can certainly be used, we used simple voting as it gave us the same performance as more complicated rules.

To give the reader an idea about the savings, the ensemble classifier can be trained on $2 \times 9,074$ images with a $10,000$-dimensional feature set with $L = 31$ and $d_{\text{red}} = 1600$ and at the same time make decisions about the entire BOSSrank in about 7 minutes. This was achieved on a DELL Precision T1500 machine with 8GB of RAM and 8 Intel Cores i7 running at 2.93GHz. The same task when approached using a G-SVM takes substantially longer. Just obtaining the performance for a *single* grid point in cross-validation took between 2–17 hours, depending on the SVM parameters. Most importantly, however, the speed and simplicity of ensemble classifiers does not seem to compromise their performance. When comparing our BOSSrank scores obtained using the ensemble classifier and G-SVMs, the values were comparable and often in favor of the ensemble classifier. We view this approach to steganalysis as a viable fully-functional and scalable alternative to SVMs.

8 The behemoths and the final attack – when 1% seems like infinity

The scalability and low-complexity of the ensemble classifier enabled us to improve our BOSSrank score simply by gradually scaling up our features and training sets. On November 15, we reached the milestone of 77% with a set consisting of 5,330 features trained with $L = 31$ and $d_{\mathrm{red}} = 1600$ on the entire BOSSbase. The set was obtained by adding the $1,458$-dimensional MINMAX vector with $T = 4$ to SUM3 (see Table 3).

On November 29, we added more features to our 5,330-dimensional set to form a feature vector with 9,288 elements. The added features were: 1) the QUANT feature vector (9) with $q = 2$ constructed from fourth-order residuals and a 4D co-occurrence (dimensionality 2×625) formed from horizontal and vertical samples as in (9) and 2) an equivalent of the QUANT feature (9) with $q = 2$ constructed from second-order residuals and a 4D co-occurrence of residuals arranged into a 2×2 square (2×625), and 3) a vector constructed from residuals computed using a translationally-invariant Ker–Böhme kernel [12]

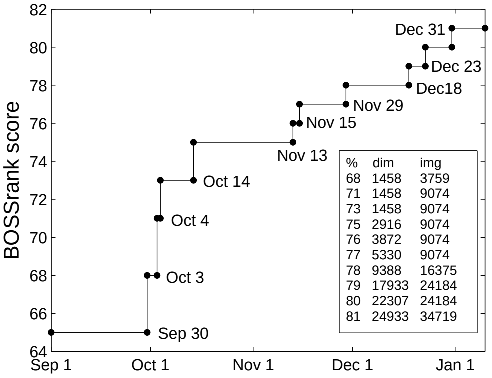
$$\begin{pmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{pmatrix}, \tag{12}$$

and a 3D co-occurrence $C^{\mathrm{h}}(\mathbf{R})+C^{\mathrm{v}}(\mathbf{R})$ (729 features) and the same co-occurrence after quantizing the residual with $q = 2$ (another 729 features). All together, the new set had $5330 + 2500 + 1458 = 9288$ features. When trained on BOSSbase, this set produced a score of 76%. However, after enlarging the training set by adding images from CAMERAS to $2 \times (9074 + 6500) = 2 \times 15,574$ images, we obtained another Hall-of-Fame entry of 78% (again with $L = 31$ and $d = 1600$ as these parameters were becoming our “sweet spot”). This submission was an eye-opener. We learned that to maximize the BOSSrank score, we had to keep a certain balance between the feature dimensionality, d, and the number of images in the training set. Given $2N$ images for training, the best results were obtained when N was by 20–50% larger than d. Training on too few images or too many would make the BOSSrank score worse. And we observed this peculiar behavior until the end of the competition. We do not have a good explanation for this oddity but hypothesize that it is one of the strange consequences of the cover-source mismatch. This rule of thumb does NOT hold when the cover-source mismatch is absent. Without the mismatch, the detection accuracy simply keeps on improving with increased feature dimensionality (see our other paper [9] in this volume).

The rest of our record submissions are displayed in Fig. 3. The last three were achieved with $L = 51, 51, 71$ and $d_{\mathrm{red}} = 2400$. The winning 24,993-dimensional feature set $\mathcal{B}$ is described in the Appendix. Our strategy was simple – keep on adding various types of features computed from different types of residuals and their quantized versions and scale the training set accordingly. We observed that the detection performance on BOSSrank was rather flat w.r.t. the parameters of the ensemble classifiers L and d_{red}. With increasing feature dimensionality, we had to increase d_{red} from 1600 to 2400 or 2800, while the number of base learners, L, did not affect the performance as much and we kept it in the range 31–81. The individual predictions converged rather fast with increased L − for the winning submission, the prediction files for BOSSrank differed in only 37 images (for $L = 31$ and 51) and in 18 images for $L = 51$ and 81.

We have also tried increasing the dimensionality up to 37,859 and the training set to $2 \times 44,138$ images but we started observing a drop in BOSSrank. This may mean that we saturated our approach but a more likely explanation is that our saturation in performance was another consequence of the cover-source mismatch.

The winning submission we selected for the final ranking reached the score of 80.3%. After the ground truth was revealed, we found out that our best prediction file had a score of 80.5%.



%	dim	img
68	1458	3759
71	1458	9074
73	1458	9074
75	2916	9074
76	3872	9074
77	5330	9074
78	9388	16375
79	17933	24184
80	22307	24184
81	24933	34719

Fig. 3. Chronological development of our BOSSrank score. The table shows the feature dimensionality and the number of cover images on which the classifier was trained. Scores 77% and larger were obtained using ensemble classifiers.

9 What have we learned?

Quite a bit. First, there is no reason why steganalysts should frown at high-dimensional feature sets. To the contrary, we believe that high-dimensional features are a necessity to attack advanced steganography. The dimensionality could probably be reduced by clever marginalization, however, automatized design using ensemble classifiers is preferable to hand-crafting the features. The ensemble classifiers offer a scalable and quite simple classification with very similar performance to that of the much more complex SVMs.

The second important lesson is the existence of the Warden's nightmare – the cover-source mismatch that manifests when a detector optimized on one source when applied on another experiences a loss of accuracy. Solving this problem appears to be extremely difficult because the mismatch can have too many forms. Just like robust statistics and robust versions of the likelihood-ratio test were developed to address the problems with robustness of optimal detectors and estimators, machine learning needs the same. Unfortunately, to the best knowledge of the authors very little appears to have been published on this important topic. If the BOSS oragizers had strictly adhered to the Kerckhoffs' principle, the cover source mismatch would never manifest and the competition would be more about breaking HUGO, which was perhaps the original motivation behind BOSS.

The steganalyst can improve the detection by training on a source with properties as close to the one from which the test images came. We tried to alleviate the negative impact of the cover-source mismatch by adding to BOSSbase all BOSSrank images after denoising (and pronouncing them as "covers") and all images after embedding in them payload of 0.4 bpp with HUGO (and pronouncing them as "stego"). The feature vectors of these 2×1000 images added to the training database should be rather close to the feature vectors of BOSSrank images, which might improve robustness to the cover source. We called this idea "training on a contaminated database" but were unable to improve our results with it. We plan to explore this rather interesting idea as part of our future effort.

10 Acknowledgements

The work on this paper was supported by Air Force Office of Scientific Research under the research grant number FA9550-08-1-0084. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of AFOSR or the U.S. Government. The authors would like to thank to the Soukal's Family, Tice Lerner, Jim Pittaresi, Thomas Gloe, Peggy Goldsberry, and Jonathan Cohen for providing their images for research purposes and to the BOSS Team for setting up the BOSS competition. The authors would also like to thank their families for their great patience and understanding during those long months while the authors intensely and endlessly pondered about HUGO. The thanks go to Monika, Nicole, and Kathy Fridrich, and Fenix Garcia Tigreros.

Appendix – the final 24,993-dimensional behemoth

For compactness, we use the following convention. Each feature set type is described using four parameters (s, q, m, T): s – the span of the difference used to compute the residual ($s = 3, 4, 5, \ldots$ for second-order residuals, third-order, etc.),

Feature type	Feature parameters (s, q, m, T)	Dimensionality
MINMAX	$(3,\{1,2\},3,3)$, $(4,\{2,3\},3,3)$, $(5,6,3,3)$	5×686
	$(3,\{1,2\},\{5,6\},1)$	$2 \times 486 + 2 \times 1458$
	$(3,2,4,2)$, $(4,\{2,3\},4,2)$, $(5,\{1,6\},4,2)$	5×1250
	$(2,\{1,2\},4,2)$	2×1250
MARKOV	$(3,\{1,2\},3,3)$	2×686
KB	$(9,\{1,2,4\},3,4)$	3×729
SQUARE	$(3,2,4,2)$	1250
CALI	$(3,2,3,3)$, $(4,\{2,3\},3,3)$	3×686
EDGE	$(6,\{1,2,4\},3,3)$	3×686

Table 4. The BOSS winner – the behemoth $\mathcal{B}$ of dimensionality 24,993.

q is the quantization step, m the order of the co-occurrence matrix, and T the truncation threshold. When a parameter is a set, the features are to be formed using all values from the set. The KB set was formed using (12) as described in Section 8. The SQUARE set is obtained from the MINMAX residual with co-occurrence elements formed by putting together residuals from 2×2 squares instead of straight lines. In the CALI set, prior to computing the features from the MINMAX residual, the image was convolved with an averaging 2×2 kernel to "erase" the embedding changes in a manner similar to calibration as proposed by Ker [11]. The residuals, $\mathbf{R}^{\text{EDGEMIN}}$ and $\mathbf{R}^{\text{EDGEMAX}}$, for EDGE were formed by taking MIN and MAX from residuals obtained using four directional kernels meant to follow edges in the image. An example of a kernel oriented along the minor-diagonal direction is:

$$\begin{pmatrix} -1 & 2 & -1 \\ 2 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}. \tag{13}$$

The final feature set for EDGE is formed as $C^{\text{h}}(\mathbf{R}^{\text{EDGEMIN}}) + C^{\text{v}}(\mathbf{R}^{\text{EDGEMIN}})$, $C^{\text{h}}(\mathbf{R}^{\text{EDGEMAX}}) + C^{\text{v}}(\mathbf{R}^{\text{EDGEMAX}})$.

References

1. P. Bas, T. Filler, and T. Pevný. Break our steganographic system – the ins and outs of organizing BOSS. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, Prague, Czech Republic, May 18–20, 2011.
2. P. Bas and T. Furon. BOWS-2. http://bows2.gipsa-lab.inpg.fr, July 2007.
3. G. Cancelli, G. Doërr, I. J. Cox, and M. Barni. A comparative study of ±1 steganalyzers. In *Proceedings IEEE International Workshop on Multimedia Signal Processing*, pages 791–796, Cairns, Australia, October 8–10, 2008.
4. C. Chen and Y.Q. Shi. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 3029–3032, May 2008.
5. T. Filler, T. Pevný, and P. Bas. BOSS. http://boss.gipsa-lab.grenoble-inp.fr/BOSSRank/, July 2010.

6. J. Fridrich. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of Lecture Notes in Computer Science, pages 67–81, Toronto, Canada, May 23–25, 2004. Springer-Verlag, New York.
7. J. Fridrich. Digital image forensic using sensor noise. *IEEE Signal Processing Magazine*, 26(2):26–37, 2009.
8. J. Fridrich and M. Goljan. On estimation of secret message length in LSB steganography in spatial domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 23–34, San Jose, CA, January 19–22, 2004.
9. J. Fridrich, J. Kodovský, M. Goljan, and V. Holub. Steganalysis of spatially-adaptive steganography. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, Prague, Czech Republic, May 18–20, 2011.
10. M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 1–13, San Jose, CA, January 16–19, 2006.
11. A. D. Ker. Steganalysis of LSB matching in grayscale images. *IEEE Signal Processing Letters*, 12(6):441–444, June 2005.
12. A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 5 1–5 17, San Jose, CA, January 27–31, 2008.
13. J. Kodovský and J. Fridrich. Calibration revisited. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 63–74, Princeton, NJ, September 7–8, 2009.
14. J. Kodovský and J. Fridrich. Steganalysis in high dimensions: Fusing classifiers built on random subspaces. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Watermarking, Security, and Forensics of Multimedia XIII*, volume 7880, pages OL 1–13, San Francisco, CA, January 23–26, 2011.
15. T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224, June 2010.
16. T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. In P. W. L. Fong, R. Böhme, and Rei Safavi-Naini, editors, *Information Hiding, 12th International Workshop*, Lecture Notes in Computer Science, pages 161–177, Calgary, Canada, June 28–30, 2010.
17. T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 3 1–3 14, San Jose, CA, January 29–February 1, 2007.
18. Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Science, pages 249–264, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.