Nonlinear Feature Normalization in Steganalysis

Mehdi Boroumand
Binghamton University
Department of ECE
Binghamton, NY 13902-6000
mboroum1@binghamton.edu

Jessica Fridrich
Binghamton University
Department of ECE
Binghamton, NY 13902-6000
fridrich@binghamton.edu

ABSTRACT

In this paper, we propose a method for normalization of rich feature sets to improve detection accuracy of simple classifiers in steganalysis. It consists of two steps: 1) replacing random subsets of empirical joint probability mass functions (co-occurrences) by their conditional probabilities and 2) applying a non-linear normalization to each element of the feature vector by forcing its marginal distribution over covers to be uniform. We call the first step random conditioning and the second step feature uniformization. When applied to maxSRMd2 features in combination with simple classifiers, we observe a gain in detection accuracy across all tested stego algorithms and payloads. For better insight, we investigate the gain for two image formats. The proposed normalization has a very low computational complexity and does not require any feedback from the stego class.

KEYWORDS

Steganography, steganalysis, machine learning, normalization, random conditioning, uniformization

ACM Reference format:
Mehdi Boroumand and Jessica Fridrich. 2017. Nonlinear Feature Normalization in Steganalysis. In *Proceedings of IH&MMSec '17, Philadelphia, PA, USA, June 20-22, 2017,* 10 pages.
https://doi.org/10.1145/3082031.3083239

1 INTRODUCTION

Currently, the most popular approach to steganalysis of digital images puts emphasis on the feature representation rather than machine learning. The so-called rich models consist of joint probability mass functions (co-occurrences) of neighboring noise residuals extracted using a large bank of both linear and non-linear filters (pixel predictors). Due to the high dimensionality of the features and the ensuing training complexity, researchers resorted to low-complexity machine learning paradigms, such as the ensemble classifier [17], its linear version [3], and regularized linear discriminants [4].

One possibility to improve the detection and better utilize the information contained in the feature vector without employing a more complex machine learning tool is to transform or preprocess the feature vector prior to classification. In [2], the authors showed that a non-linear feature transformation may enable better separation of cover and stego features with a simple decision boundary as long as the feature is a collection of co-occurrences. The approach was linked to approximating implicit feature maps in kernelized support vector machines with an explicit transformation [22, 32].

In this paper, we propose a related but different and much more simple idea based on applying a non-linear normalization to the features. It consists of two steps: L_1 normalization of random subsets of features and forcing the marginal distribution of each feature across images to be uniform. The first step is equivalent to changing the descriptor from joint distributions to conditional distributions, which is why we call it in this paper random conditioning. The second step is executed by applying the empirical cumulative density function (cdf) to each feature bin and is thus essentially a non-linear bin-dependent coordinate transformation that maximizes the entropy of each feature bin across cover images.

It is rather interesting that the proposed feature normalization leads to slightly larger gains in detection accuracy than the previously proposed explicit approximations of positive definite kernels [2]. Curiously, combining these approaches does not lead to further gain. We report the gain on four steganographic schemes embedding in the spatial domain and a wide range of payloads on two image sources – uncompressed images of BOSSbase 1.01 and its quality 85 JPEG version (decompressed JPEGs).

Our work was inspired by normalization techniques applied in convolutional neural networks conceived of to mimic inhibition schemes observed in the biological brain. In the context of machine learning, this technique is known as contrast normalization or neighborhood (local) response normalization [16, 18, 21, 26].

In the next section, we explain random conditioning and search for its single scalar parameter, the size of the random subsets. Section 3 contains description and analysis of uniformization. The proposed non-linear feature normalization is tested in Section 4, where we also discuss and interpret the results. A summary of the paper appears in Section 5.

2 FROM JOINT TO CONDITIONAL

The very first higher-order steganalysis features introduced in mid 2000's were formed as empirical Markov transition probability matrices. This applies both to the original publications on steganalysis of JPEGs [28] and spatial domain images [33] as well to the follow up work [25] and the SPAM feature [23]. The move from conditional to joint statistics (co-occurrences) came with the introduction of the embedding algorithm HUGO [24], where large third-order joint distributions of pixel differences were approximately preserved

IH&MMSec '17, June 20-22, 2017, Philadelphia, PA, USA

ACM ISBN 978-1-4503-5061-7/17/06…$15.00
https://doi.org/10.1145/3082031.3083239

by the design of the distortion function minimized in HUGO. Co-occurrences were then ported into the design of the spatial rich model [7] and its many variants [6, 8, 30, 31]. The authors of this article are not aware of any work aimed at reinvestigating the suitability of conditional probability distributions for steganalysis.

First, we briefly introduce the concept of a noise residual, its quantized form, and a joint probability distribution, the co-occurrence. For an $n_1 \times n_2$ grayscale image $x_{ij} \in \{0, \dots, 255\}$, $1 \le i \le n_1$, $1 \le j \le n_2$, let r_{ij} be a noise residual obtained by subtracting from each pixel value x_{ij} its predicted value $\hat{x}_{ij}$, $r_{ij} = x_{ij} - \hat{x}_{ij}$. Before forming co-occurrences, the residual is quantized using a quantizer $Q : \mathbb{R} \to \mathcal{Q}$ with $2T + 1$ centroids $\mathcal{Q} = \{-T, -T + 1, \dots, T\}$, $T \in \mathbb{N}$:

$$z_{ij} = Q_{\mathcal{Q}}(r_{ij}/q) \in \mathcal{Q}, \text{ for each } i, j, \tag{1}$$

where $q > 0$ is a quantization step. Typically, for 8-bit grayscale images, $q \in \{1, 1.5, 2\}$ in the SRM [7]. To curb the dimensionality of co-occurrences built from z_{ij} and to keep them well populated, small values of the threshold are typically used, such as $T = 2$.

A four-dimensional co-occurrence along the horizontal direction is a four-dimensional array $\mathbf{C} \in \mathcal{Q}^4$ defined as

$$\begin{aligned} C_{d_1d_2d_3d_4} = \frac{1}{n_1(n_2-3)} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2-3} [r_{ij} = d_1 \;\&\; r_{i,j+1} = d_2 \\ \&\; r_{i,j+2} = d_3 \;\&\; r_{i,j+3} = d_4], \end{aligned} \tag{2}$$

where $d_m \in \mathcal{Q}$, $m = 1, 2, 3, 4$ and $[P]$ is the Iverson bracket equal to 0 when statement P is true and zero otherwise. Thus, the dimensionality of $\mathbf{C}$ is $|\mathcal{Q}|^4$. For compactness, we will use vector notation for the four-dimensional indices $\mathbf{d} = (d_1d_2d_3d_4)$ belonging to $\mathcal{S} \triangleq \{(d_1, d_2, d_3, d_4)|d_m \in \mathcal{Q}\} = \mathcal{Q}^4$.

In this article, we will consider a more general approach to conditioning. Let and let $\mathcal{S}_1, \dots, \mathcal{S}_k$ be k disjoint subsets of $\mathcal{S}$ whose union is $\mathcal{S} = \cup_{l=1}^{k} \mathcal{S}_l$. For convenience, we introduce an index mapping $J : \mathcal{Q}^4 \to \{1, \dots, k\}$ that assigns to each $\mathbf{d} \in \mathcal{Q}^4$ the unique index $l \in \{1, \dots, k\}$ such that $(d_1d_2d_3d_4) \in \mathcal{S}_l$. We say that the four-dimensional array $\tilde{\mathbf{C}} \in \mathcal{Q}^4$ is obtained from $\mathbf{C}$ by conditioning on $\mathcal{S}_1, \dots, \mathcal{S}_k$ when all elements of $\tilde{\mathbf{C}}$ are obtained from $\mathbf{C}$ by

$$\begin{aligned} \tilde{C}_{\mathbf{d}} &= \Pr\{\mathbf{d}|\mathbf{d} \in \mathcal{S}_{J(\mathbf{d})}\} \\ &= \frac{C_{\mathbf{d}}}{\sum_{\mathbf{e} \in \mathcal{S}_{J(\mathbf{d})}} C_{\mathbf{e}}}, \end{aligned} \tag{3}$$

for all $\mathbf{d} \in \mathcal{Q}^4$. One can alternatively say that $\mathbf{C}$ has been L_1 normalized on $\mathcal{S}_1, \dots, \mathcal{S}_k$.

Replacing the joint distribution $\mathbf{C}$ with the conditional one $\tilde{\mathbf{C}}$ increases the contrast of bins from each $\mathcal{S}_l$, $l = 1, \dots, k$, equalizing the magnitude of the co-occurrence bins across the index sets. When the sets $\mathcal{S}_l$ are selected at random, we call this normalization *random conditioning*.

Conditioning bears strong similarity to normalization in neural networks [16, 18] applied across feature maps as implemented in, e.g., 'cuda convnets' with a local response normalization layer. The convnet documentation states that this type of normalization layer "encourages competition for big activities among nearby groups of neurons." The parallel between this layer and our conditioning becomes more clear when one considers individual co-occurrence bins as elements of feature maps that enter the normalization layer.

Table 1: Detection error of S-UNIWARD at 0.4 bpp on BOSSbase 1.01 with the non-symmetrized EDGE3x3 SRM submodel of dimensionality 625 (the last row) and its four versions conditioned on index sets of cardinality 5 and 25.

$\mathcal{S}_l$	$\lvert\mathcal{S}_l\rvert$	P_{E}
$(d_1, d_2, d_3, .)$	5	0.2851±0.0033
$(d_1, d_2, ., .)$	25	0.2829±0.0041
Random 5	5	0.2854±0.0032
Random 25	25	0.2752±0.0018
Original	625	0.2875±0.0028

To get a feeling for the effect of conditioning on steganalysis features, we start with a single SRM submodel 'EDGE3x3' (sometimes called KB submodel) on BOSSbase 1.01 [1] images with the steganographic algorithm S-UNIWARD [15] for payload 0.4 bits per pixel (bpp). We keep the feature in its non-symmetrized form, meaning its dimensionality is $5^4 = 625$ rather than 169 as in the SRM to allow for easier switching to conditional probabilities.

Table 1 shows the minimal total error probability (average of false-alarm and missed-detection rates P_{FA} and P_{D}) under equal priors

$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2}(P_{\mathrm{FA}} + P_{\mathrm{MD}}) \tag{4}$$

averaged over ten 50/50 splits of the database into training and testing sets obtained with the FLD-ensemble classifier [17] and the KB submodel conditioned on four different tessellations of all 5^4 co-occurrence indices $\mathcal{S}$. The statistical spread is the mean absolute deviation (MAD) across the ten database splits. The first two rows of the table correspond to the cases when the conditioning is performed on the first three indices $d_1d_2d_3$ and on the first two d_1d_2, respectively. Formally, for the first row, $\mathcal{S}_{d_1d_2d_3} = \{(d_1, d_2, d_3, d_4)|d_4 \in \mathcal{Q}\}$, $\mathcal{Q} = \{-2, -1, 0, 1, 2\}$, and thus $|\mathcal{S}_{d_1d_2d_3}| = 5$ for all $d_1d_2d_3$ and $\mathcal{S}_{d_1d_2} = \{(d_1, d_2, d_3, d_4)|d_3, d_4 \in \mathcal{Q}\}$ with $|\mathcal{S}_{d_1d_2}| = 25$ for the second row. The third and fourth rows correspond to $\mathcal{S}_l$ being selected uniformly at random from $\mathcal{Q}^4$. The last row is for the original KB feature vector. The conclusion that can be made from this initial experiment is that, considering the statistical spread, the transition probability matrices offer about the same detection as the joint or random conditioning on groups of five bins. Conditioning on random groups of 25, however, leads to a statistically significant improvement. Selecting the index sets $\mathcal{S}_l$ randomly seems better than in a structured manner obtained when considering the residuals as a Markov chain, which hints at the importance of diversity for the index sets. To obtain more insight, as our next experiment we forced diversity on $\mathcal{S}_l$. For the experiment, we moved to the full maxSRMd2 feature vector on BOSSbase 1.01 images for HILL and WOW embedding algorithms at 0.4 bpp while keeping the FLD-ensemble as the classifier. To prevent potential problems when conditioning on bins that are always zero, we removed from the feature all bins that are guaranteed to be zero independently of the input image (see Section 4.1 in [2] for more detail regarding the zeros in rich models). After removing the zero bins, the maxSRMd2 feature vector has a dimensionality of $D = 32,016$.

Table 2: Detection error P_E as a function of the index sets size $s = |\mathcal{S}_l|$ for HILL and WOW at 0.4 bpp with the maxSRMd2 feature when conditioning on index sets (5) with diversity forced in four different ways as explained in the text.

	HILL 0.4 bpp								
s	2	3	4	8	12	16	24	46	58
Mean	.2122±.0029	.2041±.0034	.2018±.0026	.2016±.0017	.2017±.0023	.2030±.0030	.2035±.0028	.2072±.0025	.2077±.0030
Var	.2123±.0020	.2055±.0037	.2011±.0030	.1999±.0017	.2007±.0032	.2029±.0026	.2036±.0033	.2062±.0039	.2062±.0031
σ/μ	.2067±.0018	.2035±.0029	.2021±.0029	.2008±.0024	.2003±.0013	.2033±.0027	.2029±.0024	.2061±.0029	.2077±.0019
Corr	.2106±.0025	.2043±.0025	.2027±.0026	.2018±.0040	.2013±.0021	.2016±.0029	.2030±.0031	.2060±.0021	.2056±.0027
	WOW 0.4 bpp								
Mean	.1346±.0013	.1285±.0025	.1270±.0026	.1321±.0032	.1337±.0022	.1356±.0034	.1389±.0033	.1446±.0028	.1469±.0034
Var	.1334±.0015	.1285±.0021	.1286±.0022	.1292±.0032	.1341±.0032	.1350±.0038	.1395±.0024	.1425±.0022	.1448±.0028
σ/μ	.1333±.0030	.1304±.0024	.1301±.0022	.1349±.0028	.1380±.0038	.1383±.0024	.1395±.0033	.1447±.0027	.1437±.0023
Corr	.1337±.0021	.1297±.0019	.1283±.0027	.1319±.0024	.1358±.0045	.1363±.0022	.1397±.0036	.1436±.0033	.1455±.0030

The diversity was forced on $\mathcal{S}_l$ by first ordering the features in the maxSRMd2 feature vector according to some scalar quantity and then selecting s equally spaced (interleaved) bins from the ordered feature vector. Given an integer s that divides the feature dimensionality D,

$$\mathcal{S}_l = \{l + nD/s | n = 0, \ldots, s-1\}, l = 1, \ldots, D/s. \tag{5}$$

For example, when $s = 8$ $\mathcal{S}_1$ ={1, 4003, 8005, 12007, 16009, 20011, 24013, 28015} and the last $\mathcal{S}_{4002}$ ={4002, 8004, 12006, 16008, 20010, 24012, 28014, 32016}.

We denote the ith feature (bin) in the maxSRMd2 feature vector of jth cover image as $f_i^{(j)}$, $i = 1, \ldots, D$, $j = 1, \ldots, N_{trn}$, where N_{trn} is the number of images in the training set. The following scalar quantities were investigated for ordering:

(1) Sample mean bin population across all training cover images $\mu_i = 1/N_{trn} \sum_{j=1}^{N_{trn}} f_i^{(j)}$.
(2) Sample variance of the bin $\sigma_i^2 = 1/(N_{trn}-1) \sum_{j=1}^{N_{trn}} (f_i^{(j)} - \mu_i)^2$.
(3) Relative statistical spread σ_i/μ_i.
(4) Sample correlation between bins,

$$\rho_{km} = \frac{1/N_{trn} \sum_{j=1}^{N_{trn}} (f_k^{(j)} - \mu_k)(f_m^{(j)} - \mu_m)}{\sigma_k \sigma_l}. \tag{6}$$

To obtain the ordering, all D^2 values ρ_{kl}, $1 \le k, l \le D$ are ordered from the largest to the smallest: $\rho_{k_1 l_1} \ge \rho_{k_2 l_2} \ge \rho_{k_3 l_3} \ge \ldots$. Then, the ordering is obtained as $k_1, l_1, k_2, l_2, k_3, l_3, \ldots$, while skipping over indices already present in the sequence.

Table 2 shows the detection error P_E as a function of the index subset size s for HILL [20] and WOW [11] at 0.4 bpp with the maxSRMd2 feature set. All four orderings seem to produce similar results with a minimal detection error for $4 \le s \le 8$. A simple way to force diversity is to choose the index sets $\mathcal{S}_l$ randomly, all of cardinality $s = |\mathcal{S}_l|$. Figure 1, shows the detection error $P_\mathrm{E}(s)$ and its statistical spread over ten database splits as a function of s on four steganographic algorithms and payload 0.4 bpp. Lennard–Jones potential function [19] in the form $V(x) = ax^{12} + bx^6$ was used to obtain the fit. The detection error for the original maxSRMd2 feature vector is shown on the far right to highlight the gain due to random conditioning. We note that a qualitatively similar behavior was observed for payload 0.2 bpp.

To conclude the experiments in this section, we can say that random conditioning provides approximately the same detection gain as forcing diversity with index sets (5). We choose random conditioning for the rest of this paper because this feature normalization is independent of the properties of images across the source and does not need examples of cover or stego images to estimate any parameters.

Since random conditioning contains randomness, the detection error P_E will slightly vary even when all other experimental parameters are fixed. Figure 2 shows the histogram of the detection error averaged over ten splits of the database repeated for 50 different seeds used for random conditioning. The figure was obtained for HILL at relative payload 0.2 and 0.4 bpp (left and right). We wish to point out that the distribution appears symmetrical and unimodal. The difference in P_E between the best and worst detection is approximately 0.5%. We investigated whether it is possible to identify a good seed that would consistently give good results across embedding algorithms and payloads. We could not, however, identify any consistent fluctuations. Thus, to simplify the matters, we recommend that the randomness in random conditioning be simply fixed.

3 UNIFORMIZATION

Besides conditioning as described in the previous section, the second measure we propose in this paper is normalization across images. Because a typical linear normalization would have no effect when coupled with a linear classifier, we apply a non-linear procedure that ensures that the marginal distribution of each feature j has the maximal entropy. That is, we force it to be uniform on $[0, 1]$ across images (j), $f_i^{(j)} \sim U[0, 1]$ for each bin i.

In general, given n independent realizations $x_1, \ldots, x_n$ of a random variable X sorted from the smallest to the largest in a non-decreasing sequence, the empirical cumulative density function (c.d.f.) of X is

$$F(x) = \begin{cases} \frac{l-1}{n}, l = \arg\min_l x < x_l, & \text{when } x < x_n \\ 1 & \text{when } x \ge x_n. \end{cases} \tag{7}$$

To force $f_i^{(j)} \sim U[0, 1]$ across images j for each bin i, we use the realizations $f_i^{(j)}$, $j = 1, \ldots, N_{trn}$, to estimate the empirical c.d.f. $F_i(x)$ using Eq. (7). Because this normalization is a property

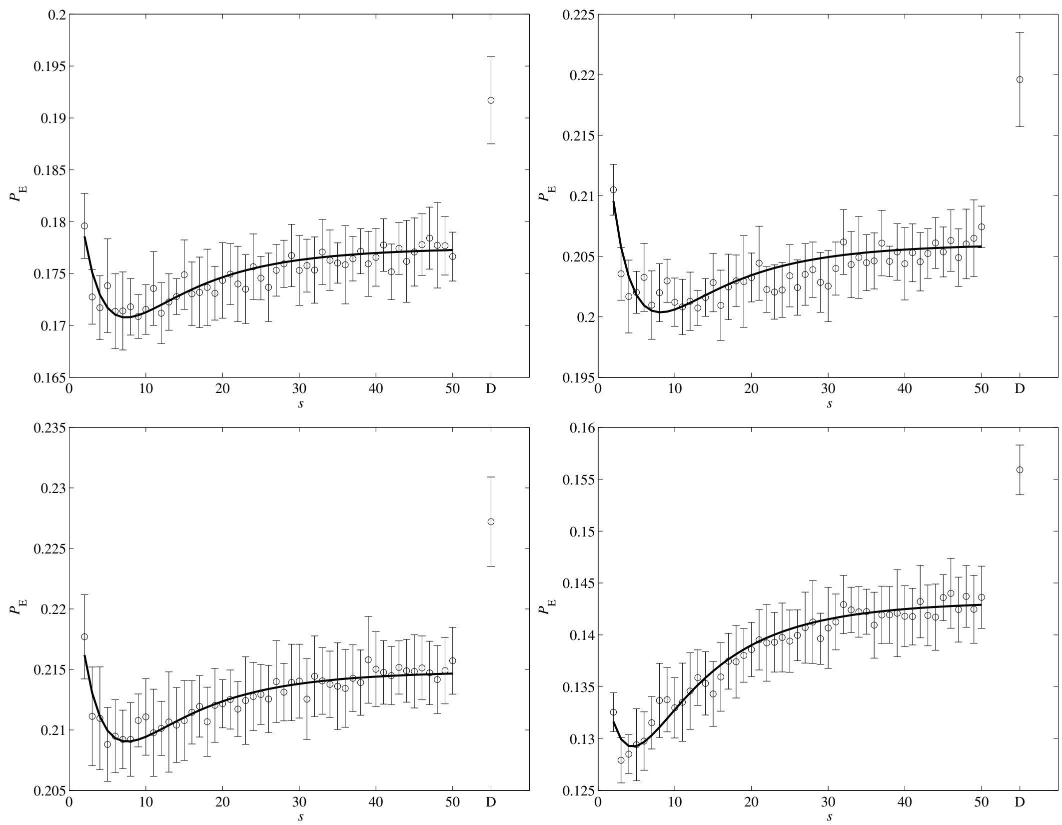

Figure 1: Detection error $P_E(s)$ as a function of the random set size $s = |\mathcal{S}_I|$. The last datapoint corresponds to $s = D$, the full feature dimensionality (no conditioning). Left to right, top to bottom: S-UNIWARD, HILL, MiPOD, WOW, payload 0.4 bpp, BOSSbase 1.01, maxSRMd2.

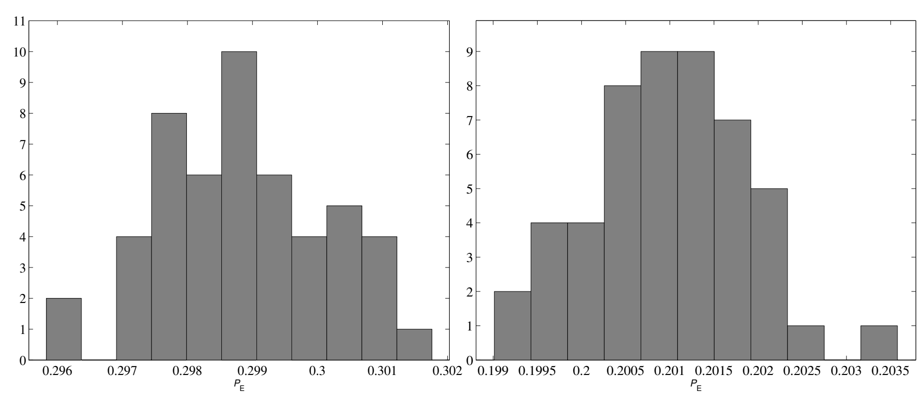

Figure 2: Histogram of the average detection error P_E across 50 seeds used for random conditioning with $s = 8$ for HILL on BOSSbase 1.01 using maxSRMd2. Left: payload 0.2 bpp, Right: payload 0.4 bpp.

Table 3: Detection error P_E for HILL and WOW at 0.4 bpp with the maxSRMd2 feature when applying the uniformization to all bins (row 2), combing uniformization on all bins with random conditioning (RC), and combining uniformization on selected bins coupled with random conditioning (rows 4–7).

	Normalization	HILL	WOW
1	Original	0.2196±0.0039	0.1559±0.0024
2	Uniform	0.2072±0.0031	0.1349±0.0025
3	RC only	0.2008±0.0030	0.1295±0.0025
4	32,016 + RC	0.1995±0.0028	0.1263±0.0025
5	20,000 + RC	0.1972±0.0027	0.1255±0.0022
6	15,000 + RC	0.1987±0.0029	0.1243±0.0025
7	10,000 + RC	0.1996±0.0030	0.1248±0.0032
8	5,000 + RC	0.1989±0.0031	0.1257±0.0022

of the source, it needs a training set of cover images from which the empirical c.d.f. is estimated.

To observe the effect of uniformization, we selected two embedding algorithms, HILL and WOW, and payload 0.4 bpp on BOSSbase. All results appear in Table 3, which we now comment upon. The first four rows show the detection error for the original maxSRMd2 feature vector after applying uniformization to all bins, applying only random conditioning (RC), and combining uniformization with random conditioning. The parameter s for RC was chosen $s = 4$ for WOW and $s = 8$ for HILL, respectively. Comparing the effect of RC with uniformization (row 3 and 2) to the original feature (row 1), one can conclude that while both measures boost the detection, the RC has a more beneficial effect. Also, an additional small gain is obtained when combining them (row 4).

The marginal distribution of the individual bins in the maxSRMd2 feature vector varies greatly. Figure 3 shows four examples of such distributions (left column) together with the impact of embedding on the bin (right column) in the form of graphs showing the bin population after embedding versus before embedding (stego vs. cover bin population). The diagonal line should help the reader infer the impact of embedding on the bin population. Notice the scale of the x axis, which informs us about the typical population of the bin across images. The embedding has a strong impact on the bin shown in the top graph, only a rather small impact on the next two bins, and virtually no impact on the fourth bin at the bottom of the figure. Generally speaking, we noticed that all bins whose marginal distribution is similar to what is shown in the first graph are affected by embedding the most. One can also say that the bins with marginal distribution similar to the first bin correspond to the most populated and most correlated bins from the feature vector. Based on extensive experiments, we determined that such bins benefit from being non-linearly normalized (uniformized) while it is beneficial to not apply such a normalization to the remaining bins.

Based on this finding, we adjusted the uniformization to be applied only to the first w bins when ordering them according to their correlation as explained in the previous section. Rows 4–8 contain the detection error when the maxSRMd2 feature is first randomly conditioned and then the first $w \in \{D, 20000, 15000, 10000, 5000\}$ bins uniformized with the remaining $D - w$ bins left untouched. A further small gain seems to be obtained when applying the uniformization only to the first $w \approx D/2$ bins when sorting them based on correlation. This finding is consistent with what was observed for other embedding algorithms, payloads, and across sources.

In general, we found it rather difficult to optimize the non-linear coordinate normalization by trying to find alternative ways to selectively normalize. In fact, if the individual bins were independent, the log-likelihood ratio in its empirical form learned (estimated) from the training set would be an optimal "normalization" or, more properly, statistical test for steganalysis. However, in the presence of complex non-linear dependencies among individual bins, we were forced to resort to heuristics.

Even though the selective uniformization is unlikely to be close to an optimal way of normalizing the bins, it is beneficial as it lowers the detection error and decreases the computational complexity.

4 EXPERIMENTS

In this section, we experimentally evaluate the proposed feature normalization on four steganographic algorithms, five payloads, and two cover sources - BOSSbase 1.01 and BOSSbaseJ85. BOSSbaseJ85 (J as in JPEG, 85 is the JPEG quality factor) was formed from BOSSbase 1.01 images by JPEG compressing them with quality factor 85 and then decompressing to the spatial domain and representing the resulting image as an 8-bit grayscale. The low-pass character of JPEG compression makes the images less textured and much less noisy. The tested steganographic schemes include MiPOD [27], HILL [20], S-UNIWARD [15], and WOW [11].

Before we present the results of the detection, we provide a pseudo-code for the experimental routine to clarify the procedure that was applied to the features before classification.

Algorithm 1 Training a classifier with N_{trn} training images by normalizing with D-dimensional cover/stego features stored as matrices $\mathbf{f}^{(c)} \in \mathbb{R}^{N_{trn}\times D}$ and $\mathbf{f}^{(s)} \in \mathbb{R}^{N_{trn}\times D}$. The same random conditioning with permutation P is done to features from the test set. The uniformization learned on the training set (the permutation R and $F_{R(i)}$, $i = 1, \ldots, D/2$) is then also applied to all features from the testing set.

Set set size for RC
Generate random permutation P of indices $1, \ldots, D$
Apply random conditioning to each row of $\mathbf{f}$:
for $l = 1, \ldots, D/s$ **do**
    for $j = 1 : N_{trn}$ **do**
        $\mathbf{f}^{c/s}(j, P((l-1)s+1 : ls) \leftarrow \frac{\mathbf{f}^{c/s}(j,P((l-1)s+1:ls))}{\sum_{k=(l-1)s+1}^{ls}\mathbf{f}^{c/s}(j,k)}$
    end for
end for
Order all D cover features by correlation (Eq. (6)), denote order R (a permutation of $1, \ldots, D$)
for $i = 1, \ldots, D/2$ **do**
    Compute $F_{R(i)}$ (Eq. (7)) for N_{trn} samples $\mathbf{f}^c(:, R(i))$
    for $j = 1 : N_{trn}$ **do**
        Apply $F_{R(i)}$ to $\mathbf{f}^c(j, R(i))$ and $\mathbf{f}^s(j, R(i))$
    end for
end for

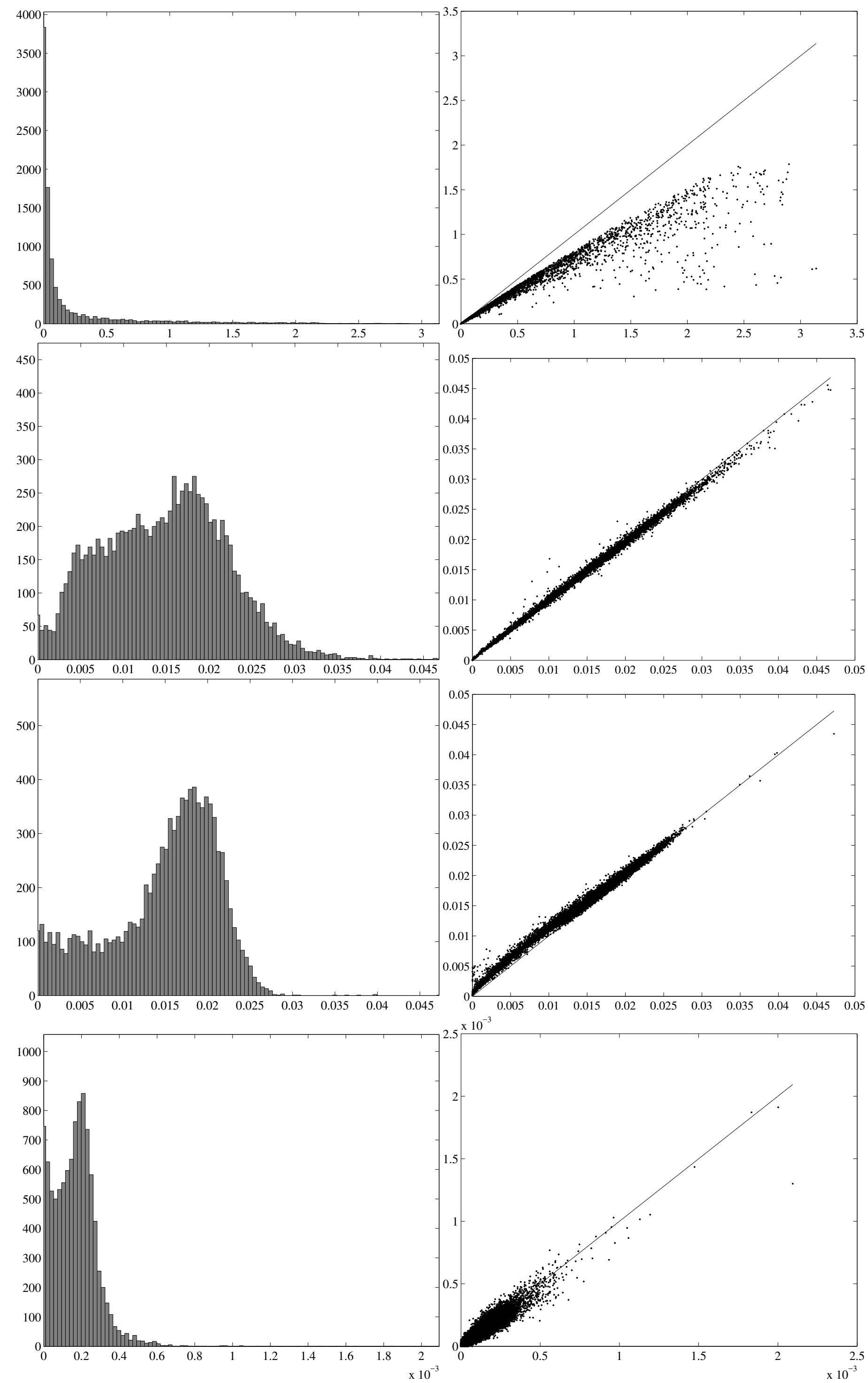

Figure 3: Examples of marginal (cover) distributions of four bins (left) from maxSRMd2 feature vector and the impact of embedding on the bin by plotting the cover bin population vs. stego bin population (right). The graphics was obtained across the entire BOSSbase database for HILL at 0.4 bpp. The bin indices are 16054, 24327, 19107, and 23974 in the maxSRMd2 feature after removing all zero bins.

Table 4: Detection error $\overline{P}_{\mathrm{E}}$ for four steganographic schemes and five payloads in bpp on BOSSbase 1.01 with the FLD-ensemble trained with maxSRMd2 features.

	Payload (bits per pixel)				
S-UNI	0.1	0.2	0.3	0.4	0.5
maxSRMd2	0.3652±0.0008	0.2919±0.0023	0.2374±0.0023	0.1917±0.0042	0.1569±0.0035
Square root	0.3588±0.0025	0.2851±0.0034	0.2276±0.0021	0.1785±0.0033	0.1433±0.0026
exp-Hellinger	0.3608±0.0033	0.2803±0.0027	0.2181±0.0028	0.1720±0.0020	0.1348±0.0025
RC	0.3614±0.0030	0.2818±0.0026	0.2190±0.0028	0.1721±0.0034	0.1334±0.0030
RC+SU	0.3618±0.0020	0.2788±0.0014	0.2156±0.0023	0.1701±0.0035	0.1307±0.0032
HILL					
maxSRMd2	0.3742±0.0022	0.3105±0.0033	0.2580±0.0033	0.2196±0.0039	0.1815±0.0033
Square root	0.3669±0.0032	0.3007±0.0025	0.2512±0.0036	0.2116±0.0026	0.1736±0.0030
exp-Hellinger	0.3653±0.0024	0.2974±0.0028	0.2451±0.0024	0.2004±0.0019	0.1649±0.0031
RC	0.3661±0.0030	0.2998±0.0024	0.2453±0.0030	0.2031±0.0044	0.1655±0.0039
RC+SU	0.3655±0.0020	0.2980±0.0014	0.2408±0.0022	0.2008±0.0022	0.1627±0.0020
MiPOD					
maxSRMd2	0.3949±0.0031	0.3246±0.0034	0.2709±0.0027	0.2272±0.0037	0.1865±0.0029
Square root	0.3926±0.0047	0.3185±0.0022	0.2635±0.0027	0.2209±0.0036	0.1818±0.0022
exp-Hellinger	0.3911±0.0038	0.3148±0.0026	0.2568±0.0024	0.2104±0.0028	0.1720±0.0031
RC	0.3903±0.0037	0.3115±0.0027	0.2541±0.0021	0.2112±0.0044	0.1733±0.0032
RC+SU	0.3900±0.0029	0.3111±0.0032	0.2516±0.0046	0.2068±0.0030	0.1690±0.0033
WOW					
maxSRMd2	0.2984±0.0020	0.2331±0.0018	0.1907±0.0028	0.1559±0.0024	0.1279±0.0030
Square root	0.2854±0.0033	0.2140±0.0031	0.1702±0.0026	0.1375±0.0020	0.1118±0.0033
exp-Hellinger	0.2820±0.0024	0.2094±0.0025	0.1645±0.0031	0.1310±0.0028	0.1068±0.0032
RC	0.2826±0.0040	0.2113±0.0027	0.1633±0.0039	0.1301±0.0035	0.1055±0.0019
RC+SU	0.2801±0.0032	0.2051±0.0019	0.1588±0.0023	0.1257±0.0036	0.1017±0.0024

We note that the permutation P of indices $\{1,\ldots,D\}$ for random conditioning is generated and then fixed across all experiments. The feature order R by correlation (6) and the c.d.f.s $F_{R(i)}$, $i = 1,\ldots,D/2$, are learned from all N_{trn} cover features from the training set and then applied to the testing set. The size of the random subsets is set to four for WOW and eight for other embedding schemes. The results of experiments on BOSSbase 1.01 and BOSSbaseJ85 are reported in Tables 4 and 5, respectively. As above, random conditioning is abbreviated as RC and, when combined with selective uniformization, we abbreviate as RC+SU. The results are also contrasted with what can be achieved with preprocessing the features using explicit non-linear maps [2]. Note that in most cases random conditioning achieves the same performance as the transformation with the exponential Hellinger kernel. As explained in the previous section, due to the randomness in RC, the results for RC can be slightly better or worse depending upon which seed is used for the random permutation. In our experiments, we fixed our seed ('seed = 1' in Matlab's Mersenne twister generator) for all tested steganographic methods, payloads, and image sources.

While combining random conditioning with selective uniformization further improves the detection performance, the improvement due to random conditioning is much larger than that of selective uniformization. The detection accuracy can be enhanced by up to 2.5% using random conditioning and up to 0.6% additional improvement can be achieved using selective uniformization. The effect of selective uniformization is most pronounced for WOW.

Since BOSSbaseJ85 is less noisy than BOSSbase 1.01, it is easier to steganalyze thus the detection error rates are overall much lower. While a consistent gain is observed for random conditioning, selective uniformization generally does not help for this source.

Figure 4 shows a graphical representation of how the proposed normalization affects the detection performance of maxSRMd2 for all tested embedding methods at two payloads, 0.2 bpp and 0.4 bpp, for both image sources. Normalization generally helps more for larger payloads than for smaller payloads. As already mentioned above, selective uniformization does not bring any performance boost in BOSSbaseJ85. Its effect also fades at the lower payloads for BOSSbase.

Finally, we note that, similar to the previously proposed explicit non-linear mappings of features, random conditioning and selective uniformization do not improve performance of features formed by histograms of residuals, such as the projection spatial rich model [12] and JPEG-phase-aware features [5, 13, 14, 29] for detection of modern JPEG steganography [9, 10, 15]. This is likely due to the fact that the bins of such feature vectors are better populated with far smaller differences between the least and most populated

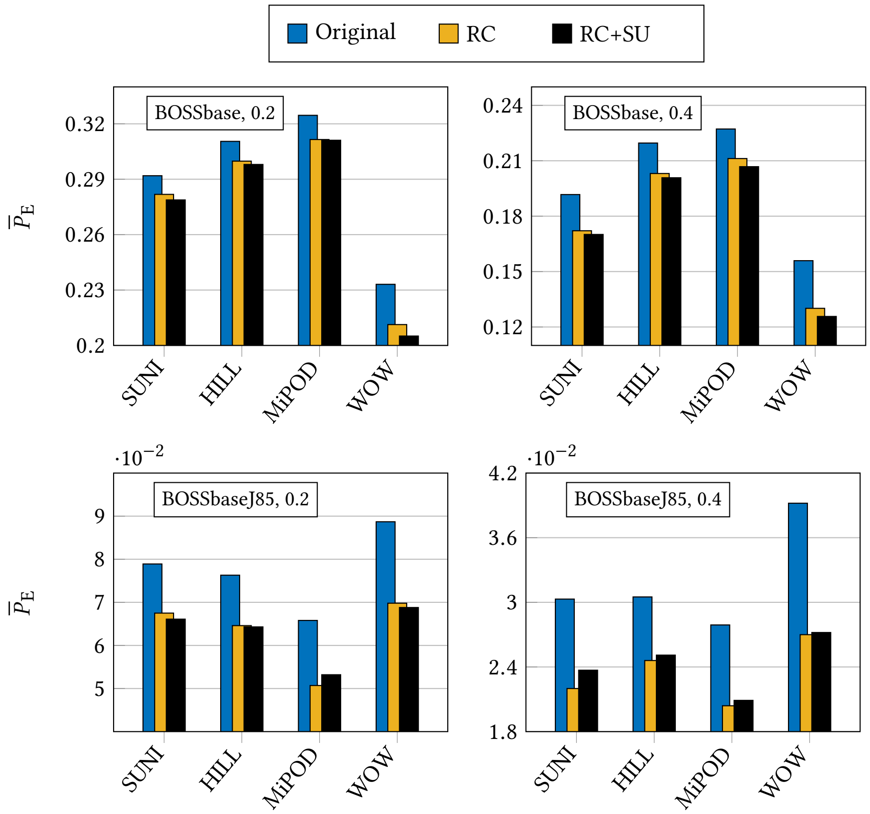

Figure 4: $\overline{P}_E$ for four different embedding schemes and two image sources at 0.2 bpp and 0.4 bpp with the FLD-ensemble trained with maxSRMd2 feature set and its normalized versions.

bins. With a more uniform distribution of the bins across images, the normalization methods proposed here are naturally less likely to be effective.

5 CONCLUSION

In this paper, we propose a low-complexity method for feature normalization of rich feature sets built as co-occurrences to improve the detection performance of simple classifiers. It adds only negligible computational overhead to feature computation and can be considered as a cheap pre-processing step before feeding the feature sets to a classifier.

We introduced two types of normalization: normalization on random subsets of the feature set called random conditioning and normalization of each bin across the database, uniformization. Random conditioning can be interpreted as switching from a joint distribution to a conditional distribution. It does not require any training data and can be applied to feature sets independently of the cover source, embedding algorithm, and payload. Since the inherent randomness associated with this process causes fluctuations in the final detection rate by approximately ±0.5% in terms of P_E, the authors encourage researchers employing this normalization method to specify the seed used for generating the random subsets in their papers.

Experimental results show a consistent performance improvement across all tested steganographic methods, payloads, and databases. Random conditioning is more effective than selective uniformization and is responsible for most of the gain we observed. In particular, in decompressed JPEGs, selective uniformization was observed as ineffective.

6 ACKNOWLEDGMENTS

The work on this paper was supported by Air Force Office of Scientific Research under the research grant number FA9950-12-1-0124. 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 anonymous reviewers for their insightful comments.

REFERENCES

[1] P. Bas, T. Filler, and T. Pevný. 2011. Break Our Steganographic System – the Ins and Outs of Organizing BOSS. In *Information Hiding, 13th International Conference* (Lecture Notes in Computer Science), T. Filler, T. Pevný, A. Ker, and S. Craver (Eds.), Vol. 6958. Prague, Czech Republic, 59–70.

Table 5: Detection error $\overline{P}_{\mathrm{E}}$ for four steganographic schemes and five payloads in bpp on BOSSbaseJ85 with the FLD-ensemble trained with maxSRMd2 features.

	Payload (bits per pixel)				
S-UNI	0.1	0.2	0.3	0.4	0.5
maxSRMd2	0.1527±0.0019	0.0789±0.0016	0.0470±0.0018	0.0303±0.0013	0.0189±0.0011
Square root	0.1410±0.0016	0.0698±0.0018	0.0404±0.0012	0.0253±0.0015	0.0164±0.0012
exp-Hellinger	0.1404±0.0020	0.0691±0.0017	0.0402±0.0018	0.0241±0.0009	0.0147±0.0011
RC	0.1381±0.0018	0.0675±0.0021	0.0373±0.0006	0.0220±0.0014	0.0133±0.0009
RC+SU	0.1355±0.0024	0.0661±0.0020	0.0384±0.0016	0.0237±0.0015	0.0143±0.0007
HILL					
maxSRMd2	0.1404±0.0012	0.0763±0.0020	0.0474±0.0024	0.0305±0.0011	0.0213±0.0011
Square root	0.1311±0.0019	0.0697±0.0027	0.0407±0.0016	0.0271±0.0011	0.0188±0.0015
exp-Hellinger	0.1284±0.0014	0.0670±0.0023	0.0390±0.0020	0.0257±0.0013	0.0172±0.0009
RC	0.1235±0.0019	0.0646±0.0019	0.0378±0.0018	0.0246±0.0017	0.0158±0.0008
RC+SU	0.1241±0.0017	0.0643±0.0017	0.0383±0.0013	0.0251±0.0011	0.0159±0.0010
MiPOD					
maxSRMd2	0.1191±0.0016	0.0658±0.0023	0.0416±0.0023	0.0279±0.0016	0.0203±0.0008
Square root	0.1135±0.0024	0.0627±0.0021	0.0395±0.0021	0.0280±0.0020	0.0190±0.0007
exp-Hellinger	0.1083±0.0024	0.0555±0.0014	0.0344±0.0021	0.0228±0.0016	0.0161±0.0010
RC	0.1038±0.0020	0.0507±0.0030	0.0312±0.0016	0.0204±0.0013	0.0136±0.0008
RC+SU	0.1061±0.0026	0.0532±0.0025	0.0326±0.0007	0.0209±0.0008	0.0147±0.0008
WOW					
maxSRMd2	0.1599±0.0021	0.0887±0.0027	0.0582±0.0026	0.0392±0.0019	0.0262±0.0016
Square root	0.1452±0.0026	0.0783±0.0020	0.0499±0.0018	0.0325±0.0016	0.0223±0.0020
exp-Hellinger	0.1398±0.0012	0.0755±0.0025	0.0468±0.0014	0.0304±0.0012	0.0198±0.0012
RC	0.1383±0.0023	0.0698±0.0015	0.0438±0.0015	0.0270±0.0012	0.0172±0.0013
RC+SU	0.1332±0.0017	0.0688±0.0019	0.0427±0.0018	0.0272±0.0017	0.0179±0.0011

[2] M. Boroumand and J. Fridrich. 2016. Boosting Steganalysis with Explicit Feature Maps. In *4th ACM IH&MMSec. Workshop*, F. Perez-Gonzales, F. Cayre, and P. Bas (Eds.). Vigo, Spain.

[3] R. Cogranne and J. Fridrich. 2015. Modeling and Extending the Ensemble Classifier for Steganalysis of Digital Images Using Hypothesis Testing Theory. *IEEE Transactions on Information Forensics and Security* 10, 2 (December 2015), 2627–2642.

[4] R. Cogranne, V. Sedighi, T. Pevný, and J. Fridrich. 2015. Is Ensemble Classifier Needed for Steganalysis in High-Dimensional Feature Spaces?. In *IEEE International Workshop on Information Forensics and Security*. Rome, Italy.

[5] T. Denemark, M. Boroumand, and J. Fridrich. 2016. Steganalysis Features for Content-Adaptive JPEG Steganography. *IEEE Transactions on Information Forensics and Security* 11, 8 (Aug 2016), 1736–1746.

[6] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich. 2014. Selection-Channel-Aware Rich Model for Steganalysis of Digital Images. In *IEEE International Workshop on Information Forensics and Security*. Atlanta, GA.

[7] J. Fridrich and J. Kodovský. 2011. Rich Models for Steganalysis of Digital Images. *IEEE Transactions on Information Forensics and Security* 7, 3 (June 2011), 868–882.

[8] M. Goljan, R. Cogranne, and J. Fridrich. 2014. Rich Model for Steganalysis of Color Images. In *Sixth IEEE International Workshop on Information Forensics and Security*. Atlanta, GA.

[9] L. Guo, J. Ni, and Y.-Q. Shi. 2012. An Efficient JPEG Steganographic Scheme Using Uniform Embedding. In *Fourth IEEE International Workshop on Information Forensics and Security*. Tenerife, Spain.

[10] L. Guo, J. Ni, and Y. Q. Shi. 2014. Uniform Embedding for Efficient JPEG Steganography. *IEEE Transactions on Information Forensics and Security* 9, 5 (2014).

[11] V. Holub and J. Fridrich. 2012. Designing Steganographic Distortion Using Directional Filters. In *Fourth IEEE International Workshop on Information Forensics and Security*. Tenerife, Spain.

[12] V. Holub and J. Fridrich. 2013. Random Projections of Residuals for Digital Image Steganalysis. *IEEE Transactions on Information Forensics and Security* 8, 12 (December 2013), 1996–2006.

[13] V. Holub and J. Fridrich. 2015. Low-Complexity Features for JPEG Steganalysis Using Undecimated DCT. *IEEE Transactions on Information Forensics and Security* 10, 2 (Feb 2015), 219–228.

[14] V. Holub and J. Fridrich. 2015. Phase-Aware Projection Model for Steganalysis of JPEG Images. In *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, A. Alattar and N. D. Memon (Eds.), Vol. 9409. San Francisco, CA.

[15] V. Holub, J. Fridrich, and T. Denemark. 2014. Universal Distortion Design for Steganography in an Arbitrary Domain. *EURASIP Journal on Information Security, Special Issue on Revised Selected Papers of the 1st ACM IH and MMS Workshop* 2014:1 (2014).

[16] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. 2009. What is the best Multi-Stage Architecture for Object Recognition?. In *2009 IEEE 12th International Conference on Computer Vision*. Kyoto, Japan, 2146–2153.

[17] J. Kodovský, J. Fridrich, and V. Holub. 2012. Ensemble Classifiers for Steganalysis of Digital Media. *IEEE Transactions on Information Forensics and Security* 7, 2 (2012), 432–444.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of Neural Information Processing Systems (NIPS)*. Lake Tahoe, Nevada.

[19] J. E. Lennard-Jones. 1924. On the Determination of Molecular Fields. *Proc. R. Soc. Lond. A* 106, 738 (1924), 463–477.

[20] B. Li, M. Wang, and J. Huang. 2014. A new cost function for spatial image steganography. In *Proceedings IEEE, International Conference on Image Processing, ICIP*. Paris, France.

[21] S. Lyu and E. Simoncelli. 2008. Nonlinear image representation using divisive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[22] F. Perronnin, J. Sanchez, and Yan Liu. 2010. Large-scale image categorization with explicit data embedding. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. 2297–2304.

[23] T. Pevný, P. Bas, and J. Fridrich. 2010. Steganalysis by Subtractive Pixel Adjacency Matrix. *IEEE Transactions on Information Forensics and Security* 5, 2 (June 2010), 215–224.

[24] T. Pevný, T. Filler, and P. Bas. 2010. Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. In *Information Hiding, 12th International Conference (*Lecture Notes in Computer Science), R. Böhme and R. Safavi-Naini (Eds.), Vol. 6387. Springer-Verlag, New York, Calgary, Canada, 161–177.

[25] T. Pevný and J. Fridrich. 2007. Merging Markov and DCT Features for Multi-Class JPEG Steganalysis. In *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, E. J. Delp and P. W. Wong (Eds.), Vol. 6505. San Jose, CA, 3 1–14.

[26] N. Pinto, D. D. Cox, and J. J. DiCarlo. 2008. Why is real-world visual object recognition hard? *PLOS Computational Biology* (January 25 2008).

[27] V. Sedighi, R. Cogranne, and J. Fridrich. 2016. Content-Adaptive Steganography by Minimizing Statistical Detectability. *IEEE Transactions on Information Forensics and Security* 11, 2 (2016), 221–234.

[28] Y. Q. Shi, C. Chen, and W. Chen. 2006. A Markov Process Based Approach to Effective Attacking JPEG Steganography. In *Information Hiding, 8th International Workshop (*Lecture Notes in Computer Science), J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee (Eds.), Vol. 4437. Springer-Verlag, New York, Alexandria, VA, 249–264.

[29] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang. 2015. Steganalysis of Adaptive JPEG Steganography Using 2D Gabor Filters. In *3rd ACM IH&MMSec. Workshop*, P. Comesa na, J. Fridrich, and A. Alattar (Eds.). Portland, Oregon.

[30] W. Tang, H. Li, W. Luo, and J. Huang. 2014. Adaptive Steganalysis Against WOW Embedding Algorithm. In *2nd ACM IH&MMSec. Workshop*, A. Uhl, S. Katzenbeisser, R. Kwitt, and A. Piva (Eds.). Salzburg, Austria, 91–96.

[31] W. Tang, H. Li, W. Luo, and J. Huang. 2016. Adaptive Steganalysis Based on Embedding Probabilities of Pixels. *IEEE Transactions on Information Forensics and Security* 11, 4 (April 2016), 734–745.

[32] A. Vedaldi and A. Zisserman. 2012. Efficient Additive Kernels via Explicit Feature Maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34, 3 (March 2012), 480–492.

[33] D. Zou, Y. Q. Shi, W. Su, and G. Xuan. 2006. Steganalysis based on Markov model of thresholded prediction-error image. In *Proceedings IEEE, International Conference on Multimedia and Expo*. Toronto, Canada, 1365–1368.