Reverse JPEG Compatibility Attack

Jan Butora and Jessica Fridrich, *Fellow, IEEE*

***Abstract*—A novel steganalysis method for JPEG images is introduced that is universal in the sense that it reliably detects any type of steganography as well as small payloads. It is limited to quality factors 99 and 100. The detection statistic is formed from the rounding errors in the spatial domain after decompressing the JPEG image. The attack works whenever, during compression, the discrete cosine transform is applied to integer-valued signal. Reminiscent of the well-established JPEG compatibility steganalysis, we call the new approach the "reverse JPEG compatibility attack." While the attack is introduced and analyzed under simplifying assumptions using reasoning based on statistical signal detection, the best detection in practice is obtained with machine learning tools. Experiments on diverse datasets of both grayscale and color images, five steganographic schemes, and with a variety of JPEG compressors demonstrate the universality and applicability of this steganalysis method in practice.**

***Index Terms*—Steganography, steganalysis, JPEG, quality factor 100, reverse compatibility, rounding errors, deep learning**

I. Introduction

The term "compatibility attack" is loosely used to describe a certain type of steganalysis detectors that identify stego objects by verifying either hard or probabilistic constraints that must be satisfied by all cover objects from a certain source. Typically, such attacks are universal in the sense that they work reliably on most steganographic methods as well as for very small payloads.

The first example of such an attack was the JPEG compatibility steganalysis [10] applicable whenever spatial-domain steganography is used to embed a secret in a decompressed JPEG cover image. The stego image will still bear strong traces of the JPEG compression, allowing an attacker to estimate the quantization matrix of the JPEG cover image. Since JPEG compression with a low quality factor is a many-to-one mapping, one could either mathematically prove or at least find overwhelming statistical evidence that a given 8×8 block of pixels with steganographic modifications cannot be obtained by decompressing any 8×8 block of quantized Discrete Cosine Transform (DCT) coefficients with the estimated quantization matrix. To make this attack less susceptible to loss of accuracy due to differences between JPEG compressors in practice, alternative versions of this idea were proposed by employing feature based machine learning detectors [24], [28]. Another version of this attack deals with steganalysis of LSB replacement [2], [3].

A different type of compatibility attack for color images was described in [16], where the authors show that mere eight bins in the co-occurrence corresponding to the 'min-max41c' submodel of the Color Rich Model (CRM) [15] hold all the detection power when the cover images are developed in 'dcraw' using AHD and PPG demosaicking algorithms. These eight bins are "violator bins" that are nearly empty in cover images (this is the compatibility constraint) but get populated by steganography allowing thus construction of extremely accurate detectors.

A powerful compatibility constraint in the co-occurrence corresponding to the KB residual (SQUARE3x3 submodel) in the Spatial Rich Model (SRM) [11] was also identified in parity-aware version of the SRM in [12] for steganalysis of LSB replacement for cover sources with suppressed noise, such as decompressed JPEGs or filtered images.

The compatibility attack described in this paper only applies to JPEG images compressed with standard quantization matrices with quality 99 and 100. However, after reading Section III-E, it should be clear to the reader that this attack will work for custom quantization matrices that can loosely be described as being "close" to 99 or 100. While this may seem as a severe limitation, based on the study conducted by the creators of the recent ALASKA steganalysis competition,[1] 14% of JPEG images with standard quantization matrices uploaded to Flickr have quality 100 and an additional 4% quality 99. This popularity of high quality factors may be due to the rapid decrease of storage prices combined with increased preference of users to preserve the quality of imagery they share on social platforms. Steganographers may also intentionally opt for larger JPEG qualities to increase the embedding capacity since many freely available steganographic programs, such as Jsteg [35], OutGuess [31], F5 [37], Steghide [19], Model Based Steganography [33], and JP Hide&Seek only embed in non-zero DCT coefficients. There also appears an increased interest within the forensics community in

The work on this paper was supported by NSF grant No. 1561446 and DARPA under agreement number FA8750-16-2-0173. 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 are with the Department of Electrical and Computer Engineering, Binghamton University, NY, 13902, USA. Email: {jbutora1,fridrich}@binghamton.edu

The authors would like to thank Yassine Yousfi for help with preparing some of the datasets.

[1] https://alaska.utt.fr

studying quantization noise during recompression with high quality factors [26], [29].

After introducing notation, the basics of JPEG compression, and a few preliminaries in the next section, we explain the main idea behind the reverse JPEG compatibility attack in Section III by analyzing the rounding errors in the spatial domain after decompressing a JPEG image. A statistical hypothesis formulation of the detection problem allows studying the limitations of the attack. In Section IV, we describe three machine learning built detectors trained on rounding errors and identify the most accurate detector, which is further tested in Section V and Section VI for universality and robustness to various implementations of JPEG compression, grayscale as well as color JPEGs, in established datasets, such as the union of BOSSbase and BOWS2, and a more realistic setting on the ALASKA dataset. After discussing countermeasures steganographers could use to improve the security for quality 100 in Section VII, the paper is summarized in Section VIII.

II. Preliminaries

Boldface symbols are reserved for matrices and vectors. The symbol $'\cdot'$ is used to denote elementwise product between vectors / matrices of the same dimensions. Uniform distribution on the interval $[a, b]$ will be denoted $\mathcal{U}[a, b]$ while $\mathcal{N}(\mu, \sigma^2)$ is used for the Gaussian distribution with mean μ and variance σ^2. The operation of rounding x to an integer is the square bracket $[x]$. The set of all integers will be denoted $\mathbb{Z}$. The symbol $\triangleq$ is used whenever a new concept is defined.

A. Folded Gaussian distribution

For $X \sim \mathcal{N}(\mu, s)$ with $\mu \in \mathbb{Z}$, the rounding error $X - [X] \sim \nu(x; s)$, $-1/2 \le x < 1/2$, where

$$\nu(x;s) = \frac{1}{\sqrt{2\pi s}} \sum_{n\in\mathbb{Z}} \exp\left(-\frac{(x+n)^2}{2s}\right) \tag{1}$$

$$= \frac{1}{\sqrt{2\pi s}}\Big(e^{-\frac{x^2}{2s}} + e^{-\frac{(x-1)^2}{2s}} + e^{-\frac{(x+1)^2}{2s}} + \cdots + e^{-\frac{(x-n_0)^2}{2s}} + e^{-\frac{(x+n_0)^2}{2s}} + R(x; n_0, s)\Big) \tag{2}$$

is the Gaussian distribution "folded" to the half-open interval $[-1/2, 1/2)$. The probability of specific rounding error $e \in [-1/2, 1/2)$ is basically a sum of the Gaussian distributions with means $\ldots - 1 + e, e, 1 + e, \ldots$.

It is routine to show that the remainder $R(x; n_0, s)$ is smaller than the n_0th term divided by $e^{n_0/s} - 1$ for all $x \in [-1/2, 1/2)$:

$$R(x; n_0, s) \le \left(e^{-\frac{(x-n_0)^2}{2s}} + e^{-\frac{(x+n_0)^2}{2s}}\right)\frac{1}{e^{n_0/s}-1} \le \frac{2e^{-\frac{(n_0-1/2)^2}{2s}}}{e^{n_0/s}-1} \triangleq R_{\max}(n_0, s). \tag{3}$$

Thus, the truncated sum

$$\nu(x; s, n_0) \triangleq \frac{1}{\sqrt{2\pi s}} \sum_{n=-n_0}^{n_0} \exp\left(-\frac{(x+n)^2}{2s}\right) \tag{4}$$

approximates $\nu(x; s)$

$$\max_{x\in[-1/2,1/2)} |v(x; s) - \nu(x; s, n_0)| < \delta, \tag{5}$$

once n_0 becomes large enough to satisfy

$$\frac{1}{\sqrt{2\pi s}} R_{\max}(n_0, s) < \delta. \tag{6}$$

B. Basics of JPEG compression

JPEG compression proceeds by dividing the image into 8×8 blocks, applying the DCT to each block, dividing the DCT coefficients by quantization steps, and rounding to integers. The coefficients are then arranged in a zig-zag fashion and losslessly compressed to be written as a bitstream into the JPEG file together with a header. We first describe this process for a grayscale image.

For better readability, everywhere in this paper, i, j will be strictly used to index pixels and k, l will index DCT coefficients. The original uncompressed 8-bit grayscale image with $N_1 \times N_2$ pixels is denoted $\mathbf{x} \in \{0, 1, \ldots, 255\}^{N_1 \times N_2}$. Constraining $\mathbf{x} = (x_{ij})$ to one specific 8×8 block, we will use indices $0 \le i, j \le 7$ to index the pixels in this block. During JPEG compression, the DCT coefficients before quantization, $d_{kl} \in \mathbb{R}$, are obtained using the formula $d_{kl} = \mathrm{DCT}_{kl}(\mathbf{x}) \triangleq \sum_{i,j=0}^{7} f_{kl}^{ij} x_{ij}$, $0 \le k, l \le 7$, where

$$f_{kl}^{ij} = \frac{w_k w_l}{4} \cos\frac{\pi k(2i+1)}{16} \cos\frac{\pi l(2j+1)}{16}, \tag{7}$$

$w_0 = 1/\sqrt{2}$, $w_k = 1$ for $0 < k \le 7$ are the discrete cosines. Before applying the DCT, each pixel is adjusted by subtracting 128 from it during JPEG compression, a step we omit here since it has no effect on our analysis.

The quantized DCTs are $c_{kl} = [d_{kl}/q_{kl}]$, $c_{kl} \in \{-1024, \ldots, 1023\}$, where q_{kl} are quantization steps in a luminance quantization matrix, which is supplied in the header of the JPEG file.

During decompression, the above steps are reversed. For a block of quantized DCT coefficients c_{kl}, the corresponding block of non-rounded pixel values after decompression is $y_{ij} = \mathrm{DCT}_{ij}^{-1}(\mathbf{c} \cdot \mathbf{q}) \triangleq \sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl} c_{kl}$, $y_{ij} \in \mathbb{R}$. To obtain the final decompressed image, y_{ij} are rounded to integers and clipped to a finite dynamic range $[0, 255]$.

For color images, the *RGB* representation is typically changed to YC_rC_b (luminance, and two chrominance signals), the luminance Y is processed as above, while the chrominance signals are optionally subsampled, then transformed using DCT, and finally quantized with chrominance quantization matrices, also stored in the header of the JPEG file. For more detailed description of the JPEG format, the reader is referred to [30].

III. Analysis

The key idea behind the attack and the first item studied in this section is the statistical distribution of the rounding errors in the spatial domain when decompressing a cover JPEG image. Then, a steganalysis method is developed by testing for this known distribution.

A. Cover images

We express the decompressed block of non-rounded pixels y_{ij} in terms of the original uncompressed block x_{ij} and the rounding errors in the DCT domain, $u_{kl} \triangleq d_{kl}/q_{kl} - c_{kl}$:

$$\begin{aligned} y_{ij} &= \mathrm{DCT}_{ij}^{-1}(\mathbf{c}\cdot\mathbf{q}) \\ &= \mathrm{DCT}_{ij}^{-1}(\mathbf{d}) - \mathrm{DCT}_{ij}^{-1}(\mathbf{u}\cdot\mathbf{q}) \\ &= x_{ij} - \mathrm{DCT}_{ij}^{-1}(\mathbf{u}\cdot\mathbf{q}) \end{aligned} \tag{8}$$

where

$$\mathrm{DCT}_{ij}^{-1}(\mathbf{u}\cdot\mathbf{q}) = \sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl} u_{kl}. \tag{9}$$

Assumption A1: For further analysis, we make the following assumption regarding the rounding errors of cover images in the DCT domain:

$$u_{kl} \sim \mathcal{U}[-1/2, 1/2) \tag{10}$$

$$u_{kl} \text{ mutually independent.} \tag{11}$$

for all k, l.

From the independence of u_{kl} and the fact that $E[u_{kl}] = 0$, $Var[u_{kl}] = 1/12$ for all k, l, Lindeberg's extension of the Central Limit Theorem (CLT) implies that y_{ij} approximately follows the Gaussian distribution

$$y_{ij} \sim \mathcal{N}(x_{ij}, s_{ij}), \tag{12}$$

with variance

$$s_{ij} = \frac{1}{12}\sum_{k,l=0}^{7} (f_{kl}^{ij})^2 q_{kl}^2. \tag{13}$$

Because x_{ij} is an integer, from Eq. (8) the rounding error in the spatial domain, $e_{ij} = y_{ij} - [y_{ij}]$, follows the Gaussian distribution $\mathcal{N}(0, s_{ij})$ "folded" to $[-1/2, 1/2)$, which we denoted in Section II as $e_{ij} \sim \nu(x; s_{ij})$.

B. Stego images

We model the impact of JPEG-domain steganography as adding a random variable η_{kl} with range $\{-1, 0, 1\}$ to the quantized DCT coefficients $c_{kl} \to c_{kl} + \eta_{kl}$. Assuming $\Pr\{1\} = \Pr\{-1\} = \beta_{kl}$, values β_{kl} are the so-called change rates (the selection channel) determined by the stego scheme. Thus, the decompressed non-rounded stego image z_{ij} is

$$\begin{aligned} z_{ij} &= \mathrm{DCT}_{ij}^{-1}((\mathbf{c}+\boldsymbol{\eta})\cdot\mathbf{q}) \\ &= \mathrm{DCT}_{ij}^{-1}(\mathbf{d}) + \mathrm{DCT}_{ij}^{-1}((\boldsymbol{\eta}-\mathbf{u})\cdot\mathbf{q}) \\ &= x_{ij} - \mathrm{DCT}_{ij}^{-1}(\mathbf{u}\cdot\mathbf{q}) + \mathrm{DCT}_{ij}^{-1}(\boldsymbol{\eta}\cdot\mathbf{q}). \end{aligned} \tag{14}$$

Assumption A2. The embedding changes η_{kl} are independent of the rounding errors u_{kl} and also mutually independent. This is a reasonable assumption for steganography that does not use the rounding errors as side-information for embedding.

Employing the CLT again,

$$z_{ij} \sim \mathcal{N}(x_{ij}, s_{ij} + r_{ij}), \tag{15}$$

$$r_{ij} = \sum_{k,l=0}^{7} (f_{kl}^{ij})^2 q_{kl}^2 Var[\eta_{kl}]. \tag{16}$$

Thus, the rounding error of the decompressed stego image, $e_{ij} = z_{ij} - [z_{ij}] \sim \nu(x; s'_{ij})$ with a larger variance $s'_{ij} = s_{ij} + r_{ij}$.

For example, for J-UNIWARD [22] and UED [17], [18], $Var[\eta_{kl}] = \beta_{kl}^{+} + \beta_{kl}^{-}$, where $\beta_{kl}^{+/-}$ are the change rates for changes ± 1 from the embedding simulator or the Syndrome-Trellis Code (STC) [9].

For nsF5 [13] with change rate $\beta_{kl} = \beta$ applied to non-zero AC DCTs, $Var[\eta_{kl}] = \beta$ whenever $(k, l) \neq (0, 0)$ and $c_{kl} \neq 0$.

C. Hypothesis test

The analysis carried out in the previous two subsections allows us to formulate a statistical hypothesis test for detection of steganography using rounding errors. Given a JPEG block decompressed to the spatial domain but not rounded, z_{ij}, the steganalyst is facing the following hypothesis test for all $0 \le i, j \le 7$:

$$\mathrm{H}_0 : e_{ij} \sim \nu(x; s_{ij}) \tag{17}$$

$$\mathrm{H}_1 : e_{ij} \sim \nu(x; s_{ij} + r_{ij}),\ r_{ij} > 0. \tag{18}$$

This test is composite if r_{ij} is not known, which would be the case when detecting potentially multiple steganographic methods and / or unknown payload size. On the other hand, for detecting a known steganography and a known payload size, the selection channel is approximately available – the change rates β_{kl} can be computed from the analyzed stego image – which means that r_{ij} can also be approximately computed. Finally, notice that the pair (i, j) is called the "JPEG phase" [6], [20], [21], [34].

Assuming r_{ij} is known and $r_{ij} \ll s_{ij}$, the leading term in the log-likelihood ratio test for the simple hypothesis test (17) for a single pixel i, j with rounding error e_{ij} is an energy detector:

$$L(e_{ij}) = \log \frac{\nu(e_{ij}; s_{ij} + r_{ij})}{\nu(e_{ij}; s_{ij})} \doteq -\frac{r_{ij}}{2s_{ij}} + \frac{r_{ij}}{2s_{ij}^2} e_{ij}^2. \tag{19}$$

Next, we focus on JPEG quality 100 and then consider generalizations to lower quality factors.

D. Quality factor 100

For quality factor 100, $q_{kl} = 1$ for all k, l. Since $\sum_{k,l=0}^{7} (f_{kl}^{(i,j)})^2 = 1$ due to the orthonormality of the DCT, $\mathrm{DCT}_{ij}^{-1}(\mathbf{u}\cdot\mathbf{q}) \sim \mathcal{N}(0, 1/12)$ for all pixels i, j, and $y_{ij} - [y_{ij}] \sim \nu(x; 1/12)$, $x \in [-1/2, 1/2)$:

$$\nu(x; 1/12) = \sqrt{\frac{6}{\pi}} \sum_{n\in\mathbb{Z}} \exp\left(-6(x+n)^2\right) \tag{20}$$

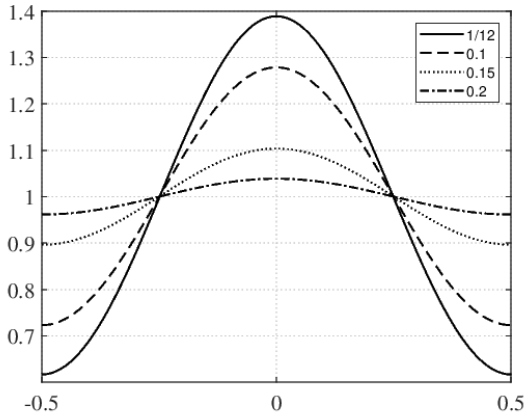

Figure 1. Distribution $\nu(x;s)$ for $s = 1/12, 0.1, 0.15, 0.2$. Note how rapidly $\nu(x;s)$ converges to a uniform distribution with increased s (also c.f. Tables I–II).

shown in Figure 1. The infinite sum is well approximated[2] with only three terms, $n \in \{-1, 0, 1\}$:

$$\nu(x;1/12) \doteq \sqrt{\frac{6}{\pi}} e^{-6x^2} \left(1 + e^{-6}(e^{12x} + e^{-12x})\right)$$
$$= \sqrt{\frac{6}{\pi}} e^{-6x^2} \left(1 + 2e^{-6}\cosh(12x)\right). \quad (21)$$

To demonstrate the performance of the energy detector (19) for this quality factor, we report the results on BOSSbase 1.01 [1] consisting of 10,000 grayscale 512×512 images compressed with Matlab's `imwrite` and embedded with the nsF5 algorithm [13] at 0.2 bpnzac (bits per non-zero AC DCT coefficient). Figure 2 left shows the distribution of the standard deviation of rounding errors in the spatial domain across all 10,000 cover and stego images while the right graph shows the ROC curve based on this test statistic. The thin right tail of the test statistic across covers gives the detector power close to 0.9 at zero false alarm. The thick left tail is due to the failure of natural images to satisfy Assumptions A1–A2. While we observed $\nu(x;1/12)$ to be a great fit to the distribution of rounding errors for most cover images, our modeling assumptions break, e.g., for images with saturated regions. Additionally, for some images the rounding error for some DCT modes fails to be uniform.

E. General quality factors

First, notice that for quality less than 100, the distribution of the inverse DCT of the rounding errors u_{kl} depends on the location i, j of the pixel in the block, its JPEG phase. Since the coefficients (7) in the DCT satisfy

$$|f_{kl}^{ij}| = |f_{kl}^{7-i,j}| = |f_{kl}^{i,7-j}| = |f_{kl}^{7-i,7-j}|, \quad (22)$$

the variance s_{ij} (13) inherits the same symmetries

$$|s_{ij}| = |s_{7-i,j}| = |s_{i,7-j}| = |s_{7-i,7-j}|. \quad (23)$$

[2]With an error less than $\delta = 2.74 \times 10^{-6}$ on the domain of ν (from (5)).

Table I
Minimum and maximum of $\nu(x,s)$ on $[-1/2, 1/2)$ as a function of variance s.

s	$\min \nu$	$\max \nu$
0.083	0.617	1.139
0.10	0.723	1.279
0.15	0.896	1.104
0.20	0.961	1.0386
0.24	0.9825	1.0175
0.30	0.9946	1.0054

Table II
Minimum and maximum variances s_{ij} over JPEG phases i, j for decreasing quality factors.

QF	$\min_{ij} s_{ij}$	$\max_{ij} s_{ij}$
100	0.083	0.083
99	0.105	0.204
98	0.2400	0.822
97	0.492	1.800
96	0.877	3.216

Thus, technically the test needs to be applied separately across 16 four-tuples of JPEG phases. However, with decreasing quality factor, the quantization steps q_{kl} increase and thus the variance s_{ij} increases as well. With increased s_{ij}, the folded Gaussian distribution $\nu(x;s_{ij})$ rapidly approaches $\mathcal{U}[-1/2, 1/2)$ (see Figure 1), which is why this steganalysis method ceases to be effective. Table I shows the minimum and maximum values of $\nu(x;s)$ on its domain $[-1/2, 1/2)$ computed for a range of variances s. Additionally, in Table II we display the minimum and maximum variance s_{ij} across JPEG phases (i, j) for decreasing quality factors.

The attack using rounding errors should still be generally effective for quality 99 because $\nu(x;s)$ is still rather far from a uniform distribution (c.f., Table I–II and Figure 1). For quality less than 99, however, $\nu(x;s)$ is so close to a uniform distribution that the attack does not work. For quality 98, this attack might still work but only when considering the rounding errors at phases $(i, j) \in \{(0,0), (0,7), (7,0), (7,7)\}$ for which the variance $s_{ij} \approx 0.24$.[3] This, however, decreases the size of available samples for the test by a factor of 16.

IV. Machine learning based detectors

Due to the complexity of natural images, Assumptions A1–A2 are satified to a varying degree, which limits the accuracy that can be achieved with detectors derived from idealized models. This motivated the authors to study machine-learning based detectors trained on the rounding errors in the spatial domain, $e_{ij} = z_{ij} - [z_{ij}]$, where z_{ij} is the decompressed but not rounded (or clipped) JPEG image. Computing the rounding errors can also be viewed as a way to suppress content and form a "noise residual."

This section describes the datasets and detectors that will be used in Section V containing the results and interpretations of all experiments.

[3]For all other phases, $s_{ij} > 0.37$, which essentially prevents the attack for typical image sizes (see Table I).

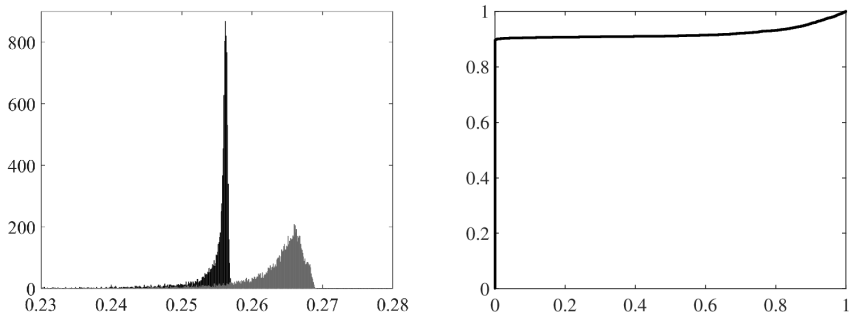

Figure 2. Left: Distribution of standard deviation of rounding errors for cover QF 100 images (black) and stego images (gray) embedded at 0.2 bpnzac with nsF5. Right: The corresponding ROC curve. Dataset: 10,000 BOSSbase grayscale 512×512 images.

A. Dataset

Two datasets were used for our experiments. The first is the union of BOSSbase 1.01 and BOWS2, each with 10,000 grayscale images resized to 256×256 pixels with `imresize` in Matlab with default parameters. This dataset is a popular choice for designing detectors with deep learning because small images are more suitable for training deep architectures [4], [38], [40]–[42], [44]. The second dataset was prepared from RAW images made available to ALASKA competitors and is detailed in Section VI-A.

B. Detectors

Three types of detectors were implemented: the SRNet [4], a deep convolutional neural network recently proposed for steganalysis in both spatial and JPEG domain, the Gabor Filter Residual features (GFR) [34] with the FLD-ensemble [25] as the classifier, and a feature set consisting of histograms of absolute values of rounding errors split by JPEG phase and symmetrized, also coupled with the ensemble classifier. Since all these detectors were trained on rounding errors e_{ij}, we abbreviate them as e-SRNet, e-GFR, and e-Hist. Next, we describe the details of each classifier and its training.

1) SRNet: For experiments on the union of BOSSbase and BOWS2, all 10,000 BOWS2 images were included in the training set, together with 4,000 randomly selected images from BOSSbase. The validation and testing set, each with 1,000 and 5,000 images were randomly selected from the remaining images from BOSSbase. The training was done for a total 25k iterations with batches of size 64 with an initial LR 2×10^{-3} that was dropped to 2×10^{-4} after 5k iterations.

2) GFR and histograms: The GFR features were extracted from the rounding errors in the spatial domain. This feature set was included as a representative of the class of JPEG phase-aware features, which are among the most powerful rich models for JPEG steganalysis.

Inspired by the analysis from Section III, we developed a third feature representation consisting of quantized histograms of absolute values of rounding errors split by JPEG phase but merged (symmetrized) across phases with the same variance (23). Formally, denoting the rounding errors in a decompressed 8×8 block of pixels $e_{ij} = z_{ij} - [z_{ij}]$, with $0 \le i, j < 7$ being the JPEG phase,

$$h_m^{(i,j)} = \left| \left\{ (i', j') \in \mathcal{P}_{ij} \middle| mq \le |e_{i'j'}| \le (m+1)q \right\} \right|, \quad (24)$$

where $q = 1/K$ is a quantization bin width with K a positive integer, $0 \le m < K/2$ the index of the histogram bin, and $\mathcal{P}_{ij}$ the set of all pixels in a $N_1 \times N_2$ image with phase (i, j): $(i', j') \in \mathcal{P}_{ij}$ if and only if $0 \le i' < N_1$, $0 \le j' < N_2$ and $\mathrm{mod}(i' - i, 8) = \mathrm{mod}(j' - j, 8) = 0$. All 64 K-dimensional histograms (24) are finally symmetrized to 16 histograms $\tilde{h}_m^{(i,j)}$ based on the symmetry of variances of rounding errors (23):

$$\tilde{h}_m^{(i,j)} = h_m^{(i,j)} + h_m^{(7-i,j)} + h_m^{(i,7-j)} + h_m^{(7-i,7-j)}, \quad (25)$$

for $0 \le i, j < 4$.

The detectors for both the GFR features and the symmetrized phase-split histograms were trained on all images not used for testing of the SRNet as described above, i.e., the training set consisted of 15,000 images for the union of BOSSbase and BOWS2. Finally, we note that $K = 10$ was used for the histograms.

V. Experiments

All experiments in this section were executed on the union of BOSSbase and BOWS2 datasets. We first show that the SRNet trained on rounding errors provides better detection than GFR or histograms on rounding errors. Further detection boost is obtained when training the SRNet on two channels – rounding errors and decompressed images, especially for QF 99. We also study the universality of the attack by showing that a detector trained on one embedding scheme can detect other, previously unseen schemes rather well as long as the SRNet is trained only on rounding errors. Finally, we investigate the robustness of this attack w.r.t. different JPEG compressors. Training on the compressor from Python's PIL generalizes overall the best.

A. Identifying the best detector

First, we studied the performance of the three machine learning detectors trained on rounding errors for quality 99

Table III
Detection accuracy of three detectors trained on rounding errors and a conventional SRNet trained on decompressed JPEGs for J-UNIWARD and a range of payloads. BOSSbase + BOWS2 dataset.

	QF 99				QF 100			
bpnzac	e-SRNet	e-GFR	e-Hist	SRNet	e-SRNet	e-GFR	e-Hist	SRNet
0.4	0.9980	0.9840	0.9376	0.8592	0.9998	0.9991	0.9933	0.8829
0.3	0.9960	0.9698	0.9035	0.8054	0.9998	0.9988	0.9865	0.8331
0.2	0.9832	0.9264	0.8376	0.7257	0.9998	0.9967	0.9702	0.7548
0.1	0.9316	0.8284	0.7218	0.6015	0.9998	0.9860	0.9212	0.6488
0.05	0.7989	0.6983	0.6239	0.5437	0.9946	0.9327	0.8486	0.5682

and 100 for J-UNIWARD and payloads 0.05–0.4 bpnzac. For comparison, in Table III we also included the results of the conventional SRNet trained on decompressed JPEG images without rounding, which is the established way of training a detector for JPEG images. For quality 100 and payloads 0.2–0.4 bpnzac, the e-SRNet is only slightly better than e-GFR (this is also due to the accuracy being very close to 1). With decreasing payload, however, e-SRNet offers better accuracy than e-GFR by up to 6% for the smallest tested payload. The phase-split histograms (e-Hist) start lagging behind e-SRNet as well as e-GFR increasingly more as the payload size decreases, with the largest loss of 14.6% w.r.t. the e-SRNet for the smallest payload 0.05 bpnzac. Note that the conventional SRNet is markedly less accurate across all payloads with the loss w.r.t. e-SRNet ranging from 11% for the largest payload to 43% for the smallest payload.

For quality 99, the e-SRNet is less accurate than for quality 100 especially for smaller payloads but still detects payload 0.4 bpnzac with 99.80% accuracy. The difference between e-SRNet and e-GFR is much larger than for quality 100. Similar to the quality 100, the phase-split histograms e-Hist and the conventional SRNet are markedly worse.

For quality 98, all three detectors trained on rounding errors were essentially randomly guessing with the exception of the three largest payloads 0.2–0.4 bpnzac where the e-SRNet achieved accuracy 0.53–0.57, respectively, at which point the conventional SRNet becomes much more accurate. This rapid loss of detection power is to be expected based on the analysis from Section III.

Next, we studied whether the performance of e-SRNet can further be improved by including the decompressed (non-rounded and non-clipped) JPEG image as a second channel (eY-SRNet). Having to train twice as many parameters in the first layer, the eY-SRNet did not converge from scratch for smaller payloads. This was addressed by curriculum learning via payload by first training on the largest payload with batch size 64 for 50k iterations with LR 2×10^{-3}, which was dropped to 2×10^{-4} for 25k more iterations. This detector is then used as a seed for training detectors for smaller payloads with the larger LR for 25k iterations, followed by 25k iterations with the smaller LR.

Table IV shows a clear benefit of using the second channel for QF 99 (eY-SRNet), especially for smaller payloads. For QF 100, the comparison is not as clear because the detection accuracy of both e-SRNet and eY-SRNet is close to 100%.

B. Universality

Based on the analysis in Section III, we expect the power of the proposed reverse compatibility attack to depend mostly on the payload size and less on the specifics of the steganographic algorithm. In this section, we evaluate the ability of e-SRNet and eY-SRNet to detect steganographic algorithms on which it was not trained on.

Three embedding algorithms were intentionally selected with vastly different embedding operations: nsF5, J-UNIWARD, and Jsteg modified to pseudo-randomly spread non-coded message bits across all DCT coefficients not equal to 0 or 1. First, a detector was trained on a small payload embedded with one of the three stego schemes and then tested on the other two. The payload for each embedding method was empirically selected so that all three embedding schemes exhibit approximately the same detectability, which is not too close to 100% or a random guesser. In particular, the detector for Jsteg was trained on payload 0.01 bpnzac, nsF5 on 0.045 bpnzac, and J-UNIWARD on 0.05 bpnzac. The results are summarized in Figure 3 showing the missed-detection probability when training on Jsteg (top), nsF5 (middle), and J-UNIWARD (bottom) and testing on stego images embedded with a range of payloads.

For QF 100 (right), the two-channel eY-SRNet performed overall better than e-SRNet. All three detectors generalize to unseen embedding very well with the detector trained on J-UNIWARD being the best. For QF 99 (left), however, e-SRNet generalizes far better than the two channel eY-SRNet, indicating perhaps that it overspecializes on the trained algorithm. Similar to QF 100, the detector trained on J-UNIWARD generalizes the best and also has the smallest false-alarm rate.

C. Robustness to JPEG compressors

Since there exist many variants of JPEG compressors, which differ mainly in the implementation of the DCT and the internal number representation, the same JPEG image may decompress slightly differently depending on the exact implementation of the DCT, and the same uncompressed image may be compressed to different JPEG files. Such differences may negativelly affect the accuracy of a detector that requires a training set, especially one trained on rounding errors. In this section, we investigate

Table IV
Detection accuracy of three different versions of SRNet when training on decompressed images (SRNet), rounding errors (e-SRNet), and both (eY-SRNet). Dataset: BOSSbase + BOWS2.

	QF 99			QF 100		
Payload	SRNet	e-SRNet	eY-SRNet	SRNet	e-SRNet	eY-SRNet
0.4	0.8592	0.9980	0.9994	0.8829	0.9998	0.9995
0.3	0.8054	0.9960	0.9990	0.8331	0.9998	0.9998
0.2	0.7257	0.9832	0.9981	0.7548	0.9998	0.9993
0.1	0.6015	0.9316	0.9780	0.6488	0.9998	0.9984
0.05	0.5437	0.7989	0.9287	0.5682	0.9946	0.9992

Table V
Testing accuracy of e-SRNet trained and tested on JPEGs for all combinations of five JPEG compressors for quality 100, J-UNIWARD 0.05 bpnzac, BOSSbase + BOWS2. The last row shows the performance of eY-SRNet when training on PIL JPEGs.

e-SRNet	Tested on images				
Trained	Matlab	Convert	Int	Float	PIL
Matlab	.9946	.9786	.9953	.9754	.9949
Convert	.8104	.9962	.8103	.9963	.8102
Int	.9964	.9823	.9960	.9790	.9963
Float	.7568	.9970	.7567	.9967	.7567
PIL	**.9959**	**.9889**	**.9966**	**.9879**	**.9959**
eY-SRNet					
PIL	.9974	.9877	.9974	.9874	.9976

this issue by purposely training on JPEG images obtained with one compressor and testing on images generated by another compressor. We do so for the embedding algorithm J-UNIWARD at quality 100 and payload 0.05 bpnzac.

The following compressors were included in our test: Matlab's imwrite, Python3 library PIL (PIL), ImageMagick's Convert (Convert), Int and Float DCT compressors in libjpeg (version 6b).[4] Fast DCT compression in libjpeg has not been included in our test because it is not recommended for quality factors larger than 97 since the compression is then slower and more lossy than on smaller quality factors.[5]

Table V shows the complete confusion matrix for quality factor 100 for e-SRNet. While a loss can indeed be observed especially in the case when the detector was built with images generated by 'Float DCT' and 'Convert', the detector trained on images from Python's PIL (boldface in the table) generalized overall very well when evaluated on images from all five compressors. With PIL generalizing the best, we also include the results for the two-channel eY-SRNet trained on images compressed by PIL to verify that adding the decompressed image as a second channel does not negatively affect robustness to different JPEG compressors.

VI. Experiments on ALASKA

To see how the reverse JPEG compatibility attack performs in more realistic conditions, we include extensive experiments on the ALASKA dataset, which contains color JPEG images of variable size, a diverse cover source with a wide spectrum of processing, four different types of stego algorithms, and variable payload size.

A. Dataset

We started with 49,928 images acquired in the RAW format provided as part of the steganalysis competition ALASKA. Available from the same web site is the script for converting RAW images to JPEGs and for embedding JPEG covers with secret messages. The conversion script develops a RAW image using four different settings and applies varying amounts of sharpening, denoising, resizing, cropping, and micro-contrast enhancement. The final size of the cover image is $N_1 \times N_2$ pixels, where $N_1, N_2 \in \{512, 640, 720, 1024\}$, obtained via "smart" crop that tries to preserve the histogram of local pixel variances (see [14] detailing the smart crop).

The embedding script selects four steganographic methods: J-UNIWARD [22], UED [17], nsF5 [13], and EBS [36] without side information, with priors 0.40, 0.30, 0.15, and 0.15, respectively. The payload size is determined by the processing chain applied by the conversion script when converting the RAW image to JPEG to obtain an approximately constant statistical detectability across various processing chains and JPEG quality factors. For example, the payload is adjusted by considering the image size based on the square root law [23]. All four embedding methods were adjusted to embed in luminance and both chrominance channels as described in [7]. The reader is referred to the above-cited ALASKA web site for more information about both scripts.

B. Training

Most deep learning architectures proposed for steganalysis [27], [32], [39]–[42], [44] cannot be trained on large images because of the memory limitations of current GPUs (11 or 12 GB). For a sufficiently large minibatch size, the images are usually limited to 256×256 or 512×512 pixels. To train a version of the SRNet that can handle images of arbitrary size, such as those from the ALASKA dataset, we adopted a similar approach as in [14] in which first a "tile detector" is trained[6] as a cover-vs-all-stego classifier on 256×256 tiles and then only its Inner Product (IP) classifier layer is retrained on images

[4]http://libjpeg.sourceforge.net/

[5]Taken from libjpeg documentation https://manpages.ubuntu.com/manpages/artful/man1/cjpeg.1.html.

[6]The batches were formed with the same priors as in the ALASKA dataset (Section VI-A).

of arbitrary size. Since the input to the IP layer in SRNet are global means of 512 feature maps outputted by the last convolutional layer, the IP layer is always presented with a 512-dimensional "feature vector" independently of the image size.

The database of RAW images was split into two disjoint parts $\mathcal{T}$ and $\mathcal{I}$, with $\mathcal{T}$ consisting of 39,188 images and $\mathcal{I}$ with 10,740 images. The images from $\mathcal{T}$ were developed with the conversion script modified to output 256×256 smart crops and were used to train the tile detector. The images from $\mathcal{I}$ were processed with the conversion script to arbitrary size and were used for retraining the IP layer on arbitrary sized images. A small portion of $\mathcal{I}$ was also used for validating the tile detector as explained next.

While validating the tile detector on 256×256 tiles was giving us 100% accuracy most of the time, we did observe different performance for different checkpoints after retraining the IP layer on arbitrary size. Thus, we validated the tile detector on arbitrary sized images from $\mathcal{I}$ as this gave more meaningful feedback to select the best checkpoint. This type of validation had to be carried out on batches consisting of one cover-stego pair because in TensorFlow framework, it is not possible to put images of different sizes in one batch.

All images in $\mathcal{T}$ were used for training the tile detector. The breakup of $\mathcal{I}$ into TRN / VAL / TST and VAL for the tile detector was 3,656 / 1,500 / 4,000, and 1,584.

The tile detector training was carried out for a total 30k iterations with mini batch size 64, starting with Learning Rate (LR) 2×10^{-3}, which was dropped to 2×10^{-4} after 10k iterations. The IP layer was retrained for 20k iterations with LR 10^{-3} and batch size 800. The setting for this layer was kept the same as for the IP layer at the end of the tile detector.

C. Searching for the best detector

Since the images in ALASKA are color, our first test was aimed at investigating whether the chrominance channels help improve detection accuracy of e-SRNet trained on rounding errors. In particular, we tested a three-channel variant of the e-SRNet in which all 64 3×3 filters in the first layer were replaced with 64 $3 \times 3 \times 3$ filters applied to rounding errors of the luminance and both chrominance signals. As Table VI shows, however, the three-chanel e-SRNet gave essentially the same results as using only the luminance. This is surprising since the conventional SRNet on quality factors other than 99 and 100 greatly benefited from including the chrominance channels [43]. We hypothesize that this is due to two reasons. First, for QFs near 100 in the ALASKA dataset, the chrominance channel carries only one half of the total payload as the luminance and thus affects the distribution of the folded Gaussian to a lesser degree. Second, since the chrominance has a narrower dynamic range than luminance, the rounding error in the DCT domain is not uniform, further violating Assumption A1 (Section III-A) under which the reverse JPEG compatibility attack was derived. All remaining experiments on ALASKA were thus executed with luminance only.

The focus of the next round of exploration was to determine whether the following design choices might perhaps further improve the detector performance:

1) Supplying the non-rounded image as a second channel (eY-SRNet)
2) Training the tile detector as multi-class instead of cover-vs.-all-stego
3) Using four moments of feature maps outputted by the tile detector to better handle images of arbitrary size
4) Using MLP instead of IP layer for the arbitrary size detector.

As observed in the previous section, while the two-channel eY-SRNet performed better than e-SRNet, it was also less robust w.r.t. a stego-source mismatch, i.e., when testing on an unseen embedding algorithm. Since the ALASKA dataset contains stego images from four different embedding schemes, it can be expected that the more robust e-SRNet will give better results than eY-SRNet. This was, indeed, confirmed experimentally as shown in Table VII. This table shows the probability of correct detection of three different versions of SRNet achieved on the ALASKA dataset with stego images following their corresponding priors, on covers (this is essentially $1 - P_{\text{FA}}$), and then on each embedding algorithm. Note that the lowest detection accuracy is for nsF5, which is due to the payload scaling applied in ALASKA (nsF5 stego images have the smallest payload).

To address the second item above, we recall the results reported in [5] on steganalysis of diversified stego sources. The authors investigated several methodologies for building a detector for stego source containing images from seven different steganographic schemes in the spatial domain. In particular, training the SRNet as multi-class (but using as binary to distinguish stego images from covers) gave better results than training it as cover-vs.-all-stego. Training the e-SRNet as multi-class, including retraining the IP layer as multi-class, however, did not translate to a gain. In fact, as Table VII shows, the correct detection was lower on ALASKA and on covers with statistically insignificant improvements for J-UNIWARD and UED.

Finally, we only comment on the effect of items 3 and 4 above. Outputting the minimum, maximum, and variance of the feature maps on the tile detector's output, in addition to the global mean, did not lead to any improvement in detection performance. Neither did we observe any gains when replacing the IP layer with a MLP with one hidden layer of double the dimensionality of the output of the tile detector. In summary, the best overall detector for QFs 99 and 100 on the ALASKA dataset was the e-SRNet trained on rounding errors of luminance only with only the global means as output of the tile detector and a simple IP layer retrained on arbitrary sizes. The tile detector as well as the IP (for arbitrary size) were trained as one-vs.-all-stego

Table VI
Detection accuracy of e-SRNet on ALASKA test set when using only the rounding errors from luminance and a three-channel e-SRNet when using the rounding errors from all three channels.

QF	99	100
Luminance	0.9400	0.9900
Color	0.9375	0.9893

classifiers on minibatches formed by respecting the priors for the four stego schemes.

VII. Countermeasures

Fundamentally, the proposed reverse JPEG compatibility attack is possible because the signals entering the DCT in the JPEG compressor are integer-valued. Therefore, a countermeasure against this attack would be to not round the luminance (and chrominances for color) before applying the DCT.

To test this hypothesis, uncompressed (16-bit TIFF) color 256×256 smart crops obtained using the developing script from ALASKA were converted to monochrome images using the relationship $Y = 0.299 \times R + 0.589 \times G + 0.114 \times B$, where R, G, B stand for red, green, and blue channel, respectively, and then scaled to the 8-bit range $[0, 255]$ without rounding. Each image was then processed using block DCT (implemented with `dct2` in Matlab). The resulting DCT coefficients were then quantized with quality 100, rounded, and finally written to a JPEG file using the JPEG Toolbox. These cover JPEGs were then embedded with J-UNIWARD at 0.2 bpnzac. With the same breakup of ALASKA into training, validation, and testing, the e-SRNet achieved the testing accuracy of 55.05, which confirms the effectiveness of this countermeasure.

This countermeasure, however, has a flaw since, to the best knowledge of the authors, all JPEG compressors round the luminance before applying the DCT. Thus, images compressed from non-rounded luminance are rare and should be suspicious by themselves. In other words, the proposed countermeasure only works within an artificially crafted cover source. In fact, since rounding errors of integer-valued compressed images follow the folded Gaussian distribution (see Figure 1) and the rounding errors of non-integer compressed images do not, both sources can be reliably distinguished: for quality 100, the SRNet tile detector trained on rounding errors of only luminance achieved 100% accuracy. Training was done with mini batch size 64 for 30k iterations, with initial LR 10^{-3} dropped to 10^{-4} after 10k iterations. The best checkpoint was selected after 4k iterations.

A more viable alternative is to break the independence of the rounding error u_{kl} and the embedding change η_{kl} (Assumption A2 in Section III-B) and ensure that the variance of $u_{kl}+\eta_{kl}$ stays as close to 1/12 as possible. This is exactly what the so-called side-informed embedding schemes [8], [18] achieve heuristically by modulating the costs of changing each DCT coefficient by $1 - 2|u_{kl}|$. Therefore, as the next step, we switched to the BOSSbase + BOWS2 dataset and tested the security of SI-UNIWARD [22] on a range of payloads for both quality factors 99 and 100 when steganalyzed with e-SRNet, eY-SRNet, and a conventional SRNet. In this case, the two-channel eY-SRNet gave overall best performance for QF 99 and for two payloads for QF 100 (see Table VIII for the complete results). Comparing the high detectability of J-UNIWARD (Table IV), we conclude that SI-UNIWARD is an effective counter measure for the reverse JPEG compatibility attack as long as the payload size is kept below 0.05 bpnzac.

VIII. Conclusions

A new compatibility steganalysis attack is proposed, which is applicable to both color and grayscale JPEG images saved with quality 99 and 100. It is based on the observation that, when decompressing a JPEG image, the rounding errors in the spatial domain exhibit a Gaussian distribution with variance 1/12 folded to $[-1/2, 1/2)$. Steganographic embedding changes made to quantized DCT coefficients increase the variance of the Gaussian distribution, allowing thus an extremely accurate detection. The attack is fundamentally possible due to the fact that the DCT is applied to integers.

While the basic principle of the attack is explained and introduced under simplifying modeling assumptions using statistical hypothesis testing, the best detectors in practice are obtained with classifiers trained on rounding errors. Three types of classifiers were investigated – Gabor Filter Residuals, phase-split histograms of rounding errors, and a deep residual network called the SRNet, which consistently provided the best results in our experiments.

The attack has been tested on five different embedding schemes, grayscale and color images, and diverse stego sources (the ALASKA dataset). It appears to be universal in the sense that a detector trained on one embedding algorithm generalizes to unseen embedding methods. The attack is also robust to various JPEG compressors. Moreover, it has been shown that steganalysis targeted to a specific embedding algorithm can be improved, especially for quality factor 99, by providing rounding errors together with decompressed image as input to the network detector.

To circumvent the attack, one needs to avoid applying the DCT to integer-valued images, which, however, none of the JPEG compressors known to the authors do. The second possibility to reduce the detectability is to use side-informed embedding schemes that minimize the combined distortion due to quantization and embedding. They, indeed, are less detectable than non-side-informed schemes. Our experiments showed that SI-UNIWARD on payload of 0.05 bpnzac essentially eluded detection. Thus, besides drastically reducing the payload, it currently appears that quality 100 and 99 JPEGs should be avoided for steganography by the same token as decompressed JPEGs should not be used for spatial-domain embedding.

Table VII
PROBABILITY OF CORRECT DETECTION OF e-SRNET, eY-SRNET, AND MULTI-CLASS e-SRNET (ALL ON LUMINANCE ONLY) ON ALASKA, COVERS ($1 - P_{\text{FA}}$), AND EACH EMBEDDING ALGORITHM. RESULTS OBTAINED ON $5 \times 4,000$ IMAGES FROM THE TEST SET.

	e-SRNet		eY-SRNet		Multi-class e-SRNet	
	QF 99	QF 100	QF99	QF100	QF 99	QF 100
ALASKA	0.9400	0.9900	0.9296	0.9450	0.9098	0.9794
Cover	0.9960	0.9985	0.9960	0.9915	0.9810	0.9993
EBS	0.9550	0.9873	0.9563	0.9788	0.9423	0.9810
JUNI	0.9945	0.9888	0.9690	0.9860	1.0000	0.9985
nsF5	0.2880	0.9508	0.2598	0.3865	0.0395	0.7945
UED	0.9825	0.9875	0.9635	0.9820	0.9910	0.9885

Table VIII
ACCURACY OF THE CONVENTIONAL SRNET TRAINED ON DECOMPRESSED IMAGES (SRNET), e-SRNET ON ROUNDING ERRORS, AND A TWO-CHANNEL eY-SRNET TRAINED ON DECOMPRESSED IMAGES AND ROUNDING ERRORS ACROSS DIFFERENT PAYLOADS OF SI-UNIWARD. DATASET: BOSSBASE + BOWS2.

	99			100		
bpnzac	SRNet	e-SRNet	eY-SRNet	SRNet	e-SRNet	eY-SRNet
0.4	0.6859	0.9517	0.9960	0.6960	0.9954	0.9941
0.3	0.6106	0.9391	0.9824	0.6186	0.9925	0.9864
0.2	0.5457	0.8354	0.9208	0.5474	0.9758	0.9915
0.1	0.5030	0.6278	0.6862	0.5474	0.8514	0.7397
0.05	0.5000	0.5110	0.5344	0.5291	0.6107	0.6800

All code used to produce the results in this paper, including the network configuration files will be made available from http://dde.binghamton.edu/download/ upon acceptance of this paper.

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 Conference*, volume 6958 of Lecture Notes in Computer Science, pages 59–70, Prague, Czech Republic, May 18–20, 2011.

[2] R. Böhme. Weighted stego-image steganalysis for JPEG covers. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lecture Notes in Computer Science, pages 178–194, Santa Barbara, CA, June 19–21, 2007. Springer-Verlag, New York.

[3] R. Böhme. *Advanced Statistical Steganalysis*. Springer-Verlag, Berlin Heidelberg, 2010.

[4] M. Boroumand, M. Chen, and J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, May 2019.

[5] J. Butora and J. Fridrich. Detection of diversified stego sources using CNNs. In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2019*, San Francisco, CA, January 14–17, 2019.

[6] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich. JPEG-phase-aware convolutional neural network for steganalysis of JPEG images. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017.

[7] R. Cogranne, Q. Giboulot, and P. Bas. The ALASKA steganalysis challenge: A first step towards steganalysis "Into the wild". In R. Cogranne and L. Verdoliva, editors, *The 7th ACM Workshop on Information Hiding and Multimedia Security*, Paris, France, July 3–5, 2019. ACM Press.

[8] T. Denemark and J. Fridrich. Side-informed steganography with additive distortion. In *IEEE International Workshop on Information Forensics and Security*, Rome, Italy, November 16–19 2015.

[9] T. Filler, J. Judas, and J. Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security*, 6(3):920–935, September 2011.

[10] J. Fridrich, M. Goljan, and R. Du. Steganalysis based on JPEG compatibility. In A. G. Tescher, editor, *Special Session on Theoretical and Practical Issues in Digital Watermarking and Data Hiding, SPIE Multimedia Systems and Applications IV*, volume 4518, pages 275–280, Denver, CO, August 20–24, 2001.

[11] J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2011.

[12] J. Fridrich and J. Kodovský. Steganalysis of LSB replacement using parity-aware features. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 31–45, Berkeley, California, May 15–18, 2012.

[13] J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.

[14] C. Fuji-Tsang and J. Fridrich. Steganalyzing images of arbitrary size with CNNs. In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2018*, San Francisco, CA, January 29–February 1, 2018.

[15] M. Goljan, R. Cogranne, and J. Fridrich. Rich model for steganalysis of color images. In *Sixth IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, December 3–5, 2014.

[16] M. Goljan and J. Fridrich. Cfa-aware features for steganalysis of color images. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[17] L. Guo, J. Ni, and Y.-Q. Shi. An efficient JPEG steganographic scheme using uniform embedding. In *Fourth IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5, 2012.

[18] L. Guo, J. Ni, and Y. Q. Shi. Uniform embedding for efficient JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 9(5):814–825, May 2014.

[19] S. Hetzl and P. Mutzel. A graph-theoretic approach to steganography. In J. Dittmann, S. Katzenbeisser, and A. Uhl, editors, *Communications and Multimedia Security, 9th IFIP TC-6 TC-11 International Conference, CMS 2005*, volume 3677 of Lecture Notes in Computer Science, pages 119–128, Salzburg, Austria, September 19–21, 2005.

[20] V. Holub and J. Fridrich. Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Transactions on*

Information Forensics and Security, 10(2):219–228, February 2015.

[21] V. Holub and J. Fridrich. Phase-aware projection model for steganalysis of JPEG images. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[22] V. Holub, J. Fridrich, and T. Denemark. 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.

[23] A. D. Ker, T. Pevný, J. Kodovský, and J. Fridrich. The Square Root Law of steganographic capacity. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 107–116, Oxford, UK, September 22–23, 2008.

[24] J. Kodovsky and J. Fridrich. JPEG-compatibility steganalysis using block-histogram of recompression artifacts. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 78–93, Berkeley, California, May 15–18, 2012.

[25] J. Kodovský, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444, April 2012.

[26] B. Li, T.-T. Ng, X. Li, S. Tan, and J. Huang. Revealing the trace of high-quality JPEG compression through quantization noise analysis. *IEEE Transactions on Information Forensics and Security*, 10(3):558–573, March 2015.

[27] B. Li, W. Wei, A. Ferreira, and S. Tan. ReST-Net: Diverse activation modules and parallel subnets-based CNN for spatial image steganalysis. *IEEE Signal Processing Letters*, 25(5):650–654, May 2018.

[28] W. Luo, Y. Wang, and J. Huang. Security analysis on spatial ± 1 steganography for JPEG decompressed images. *IEEE Signal Processing Letters*, 18(1):39–42, 2011.

[29] C. Pasquini and R. Böhme. Towards a theory of JPEG block convergence. In *IEEE International Conference on Image Processing, ICIP*, Athens, Greece, October 7–10, 2018.

[30] W. Pennebaker and J. Mitchell. *JPEG: Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, 1993.

[31] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, Washington, DC, August 13–17, 2001.

[32] Y. Qian, J. Dong, W. Wang, and T. Tan. Deep learning for steganalysis via convolutional neural networks. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[33] P. Sallee. Model-based methods for steganography and steganalysis. *International Journal of Image Graphics*, 5(1):167–190, 2005.

[34] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang. Steganalysis of adaptive JPEG steganography using 2D Gabor filters. In P. Comesana, J. Fridrich, and A. Alattar, editors, *3rd ACM IH&MMSec. Workshop*, Portland, Oregon, June 17–19, 2015.

[35] D. Upham. Steganographic algorithm JSteg. Software available at http://zooid.org/ paul/crypto/jsteg.

[36] C. Wang and J. Ni. An efficient JPEG steganographic scheme based on the block–entropy of DCT coefficents. In *Proc. of IEEE ICASSP*, Kyoto, Japan, March 25–30, 2012.

[37] A. Westfeld. High capacity despite better steganalysis (F5 – a steganographic algorithm). In I. S. Moskowitz, editor, *Information Hiding, 4th International Workshop*, volume 2137 of Lecture Notes in Computer Science, pages 289–302, Pittsburgh, PA, April 25–27, 2001. Springer-Verlag, New York.

[38] G. Xu. Deep convolutional neural network to detect J-UNIWARD. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017.

[39] G. Xu, H. Z. Wu, and Y. Q. Shi. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712, May 2016.

[40] J. Ye, J. Ni, and Y. Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, November 2017.

[41] M. Yedroudj, M. Chaumont, and F. Comby. How to augment a small learning set for improving the performances of a CNN-based steganalyzer? In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2018*, San Francisco, CA, January 29–February 1, 2018.

[42] M. Yedroudj, F. Comby, and M. Chaumont. Yedroudj-net: An efficient CNN for spatial steganalysis. In *IEEE ICASSP*, pages 2092–2096, Alberta, Canada, April 15–20, 2018.

[43] Y. Yousfi, J. Butora, Q. Giboulot, and J. Fridrich. Breaking ALASKA: Color separation for steganalysis in JPEG domain. In R. Cogranne and L. Verdoliva, editors, *The 7th ACM Workshop on Information Hiding and Multimedia Security*, Paris, France, July 3–5, 2019. ACM Press.

[44] J. Zeng, S. Tan, B. Li, and J. Huang. Large-scale JPEG image steganalysis using hybrid deep-learning framework. *IEEE Transactions on Information Forensics and Security*, 13(5):1200–1214, May 2018.

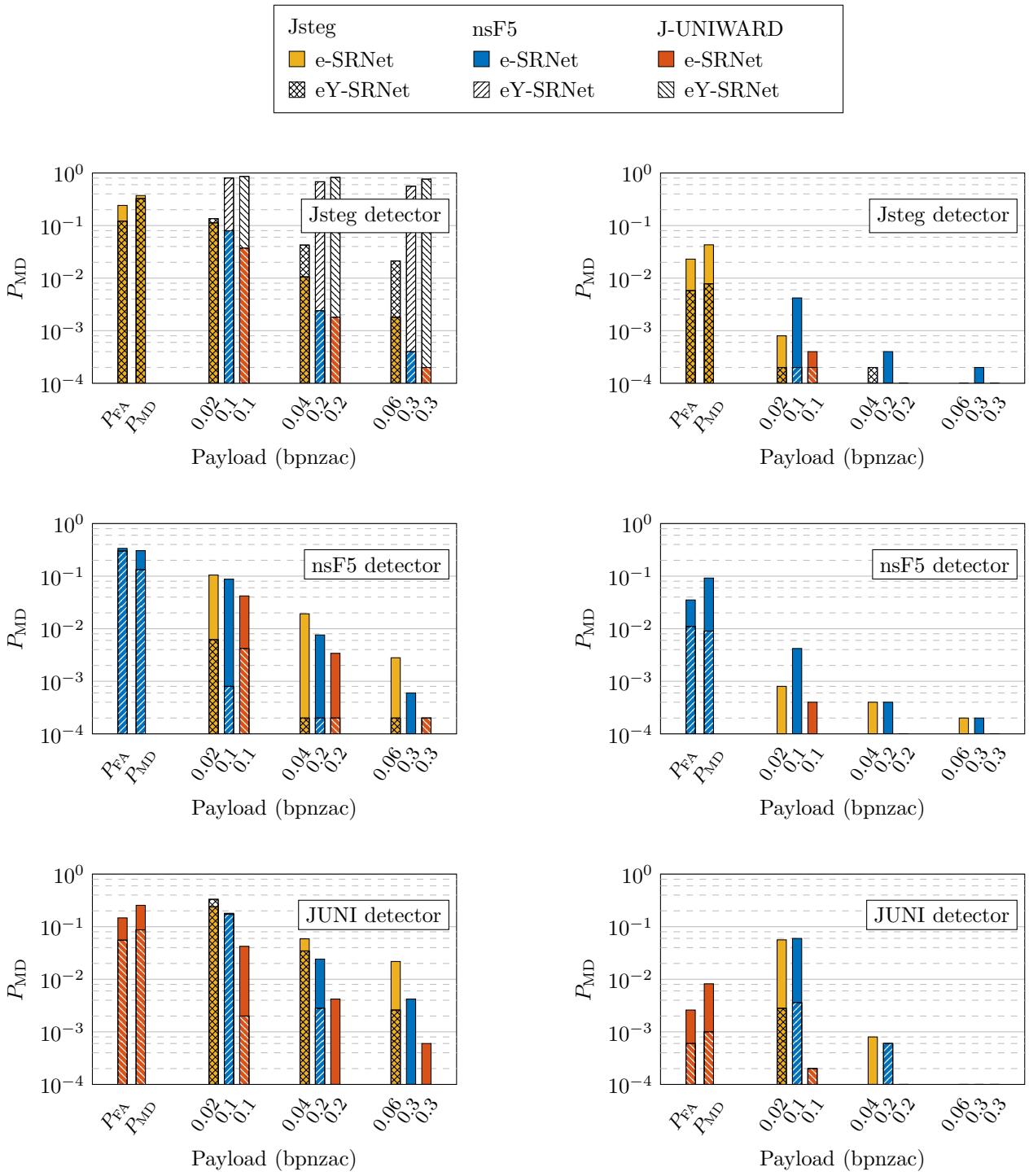

Figure 3. Probability of missed detection P_{MD} (in logarithmic scale) on stego images embedded with three different stego schemes and payloads when training e-SRNet (color) and eY-SRNet (patterns) for Jsteg (top), nsF5 (middle), and J-UNIWARD (bottom) on payloads 0.01, 0.045, and 0.05 bpnzac, respectively. The first two columns denoted by P_{FA} and P_{MD} correspond to the false-alarm and missed-detection rates of each detector. The value 10^{-4} is used to represent $P_{\text{MD}} = 0$ as this value was never achieved in terms of missed detection. Testing payloads were chosen to be roughly 2, 4 and 6 times of the payload used in training. Left: QF 99, right: QF 100. Dataset: BOSSbase + BOWS2.