Limits of Data Driven Steganography Detectors

Edgar Kaziakhmedov, Eli Dworetzky, and Jessica Fridrich
Binghamton University
Department of Electrical and Computer Engineering
Binghamton, NY 13850
{ekaziak1,edworet1,fridrich}@binghamton.edu

ABSTRACT

While deep learning has revolutionized image steganalysis in terms of performance, little is known about how much modern data driven detectors can still be improved. In this paper, we approach this difficult and currently wide open question by working with artificial but realistic looking images with a known statistical model that allows us to compute the detectability of modern content-adaptive algorithms with respect to the most powerful detectors. Multiple artificial image datasets are crafted with different levels of content complexity and noise power to assess their influence on the gap between both types of detectors. Experiments with SRNet as the heuristic detector indicate that independent noise contributes less to the performance gap than content of the same MSE. While this loss is rather small for smooth images, it can be quite large for textured images. A network trained on many realizations of a fixed textured scene will, however, recuperate most of the loss, suggesting that networks have the capacity to approximately learn the parameters of a cover source narrowed to a fixed scene.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → **Image manipulation**; **Neural networks**;

KEYWORDS

Steganography, steganalysis, deep learning, artificial source, performance limit

ACM Reference Format:
Edgar Kaziakhmedov, Eli Dworetzky, and Jessica Fridrich. 2023. Limits of Data Driven Steganography Detectors. In *Proceedings of the 2023 ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '23), June 28–30, 2023, Chicago, IL.*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3437880.3460395

1 INTRODUCTION

Modern machine learning paradigms, deep learning in particular, have predominantly been used in steganography to improve performance – to build more accurate steganography detectors [5, 7, 8, 10, 11, 22, 26, 46–48, 50, 51, 53, 54] and more secure steganographic methods [3, 21, 27, 34, 38, 39, 43–45, 55]. However, comparatively little is known about the *performance limits* of such tools and what keeps data driven detectors from reaching their best possible performance.

This lack of prior work is undoubtedly due to the formidable complexity of the task. The main difficulty is establishing the theoretical bounds of the most powerful detectors for natural images due to the lack of sufficiently accurate statistical models. In fact, establishing the exact limits may be unachievable due to the fundamental incognizability of real digital media as argued by Böhme [4].

The latest generation of detectors built as deep convolutional neural networks (CNNs) [5, 7, 22, 26, 30, 42, 46–48, 50–52] has a significant advantage with respect to the previous generation built around rich media models [14, 15, 18, 20, 32, 33, 40] and low complexity classifiers [9, 12, 24]. While rich models are essentially histograms computed from entire images and are thus macroscopic descriptors, CNNs have the ability to detect locally. In extreme cases, such detectors can reach their decision from a single, influential embedding change [49]. Given the rather sizable improvement in detection accuracy of such detectors, many researchers began asking the question of whether their theoretical limits have been achieved.

The first work that attempted to shed some light on this problem appeared in [31], where the authors used the spatial rich model (SRM) [18] as a heuristic detector, statistically independent cover pixels with the heteroscedastic ISO noise model [1, 13, 17, 19, 23, 28, 35, 36, 41], and the likelihood ratio test (LRT) as the most powerful detector. The authors concluded that in a highly homogeneous cover source consisting of multiple acquisitions of the same scene (cover source formed by a fixed scene noisified with different instances of the heteroscedastic noise), rich models performed quite close to the LRT when embedding with steganography optimal[1] to the heteroscedastic noise (MiPOD [31]). However, in a heterogeneous source consisting of different scenes, the gap between rich models and the LRT was quite large.[2] In an extension of this work [6], the authors worked with several content-adaptive embedding schemes, a range of payloads, and two types of data-driven detectors – the SRM as well as the CNN SRNet [5]. The way the data driven detectors were compared to the LRT, though, was incorrect and made the data driven detectors look better than they really were (see [16] for more detail).

The current paper builds upon the techniques developed in [6] in terms of dataset preparation but uses correct comparison of both types of detectors. Most importantly, we parametrize the formation

IH&MMSec '23, June 28–30, 2023, Chicago, IL.

ACM ISBN 978-1-4503-8295-3/21/06…$15.00
https://doi.org/10.1145/3437880.3460395

[1] Optimal in terms of the smallest deflection within the class of embedding methods that modify pixels by ±1 with equal probability.
[2] This experiment was executed with MiPOD for a detectability-limited sender.

of artificial datasets so that we can control the complexity of the content and the power of the noise present in images. On two different setups corresponding to different artificial cover sources, embedding schemes, and the type of noise, we compute the performance gap in terms of the area under the ROC curve (AUC).

In the next section, we describe our experimental setup, dataset preparation, and the embedding method investigated. This section also describes the details of the LRTs and data driven detectors and the comparison metric. In Section 3, we report the results of all our experiments and their interpretation. In Section 4, we summarize the findings, discuss the limitations of our approach, and outline future directions.

2 EXPERIMENTAL SETUP

In this paper, we consider two experimental setups schematically shown in Figure 1. Both start with a dataset of natural images that are denoised (to obtain control over content complexity) and subsequently noisified to impose a known statistical model on covers within which a closed form of the most powerful steganography detector can be derived. The setups differ in terms of the type of noise added to the denoised images and the embedding scheme studied. The first setup adds heteroscedastic noise with HILL [25] as the embedding algorithm, while the second setup adds independent Gaussian noise with variance obtained with MiPOD's [31] variance estimator with the embedding optimized to the added noise (MiPOD). These setups were selected for diversity to substantiate our conclusions.

2.1 Cover sources

All cover sources used in this study were derived from BOSSbase 1.01 [2] containing 10,000 grayscale images that were resized from their original 512×512 size to 256×256 using Matlab's `imresize` function with default parameters. This dataset is denoted as $\mathcal{B}$. The images were randomly split into training, validation, and testing sets with 4000, 1000, and 5000 images, respectively. The same split was used for all experiments in this paper. We also wish to point out that, with small changes, the generation of the artificial cover sources in this paper essentially mimics the procedure used in [6].

Since our main research objective is to determine how content complexity and noise affect the performance gap between data driven and the most powerful detectors, we created from $\mathcal{B}$ a family of 25 artificial sources with different levels of content complexity controlled by the variance of the Gaussian noise suppressed by a denoising filter and with different power of the added noise. The cover source generation is described in five steps as also illustrated in Figure 1. In a nut shell, each image in the artificial dataset was obtained by sampling from an array of independent Gaussian variables[3] $\mathcal{N}(\mu_i, \sigma_i^2)$ with μ_i obtained by denoising an image from $\mathcal{B}$ and the variance either computed from a heteroscedastic ISO noise model (Setup I) or estimated from the original image using MiPOD variance estimator (Setup II).

Step 1: Denoising. To suppress the original noise component and also to simplify (smooth) the content, all images from $\mathcal{B}$ were first denoised with the wavelet-based denoising filter with Daubechies 8-tap wavelets [29], which removes additive white Gaussian noise with standard deviation $\sigma_{\text{Den}} \in \{0.1, 0.5, 1, 3, 5\}$. This part is depicted as the 'denoise' dial in Figure 1. Depending on σ_{Den}, the resulting mean squared error (MSE) between the denoised and original images from $\mathcal{B}$ was in the range $[0, 6]$. The denoised pixels were clipped to the dynamic range $[0, 255]$, which was then further narrowed down to $[15, 240]$ by applying the following linear transform $[0, 255] \to [15, 240]$:

$$y(x) = 15 + \frac{225}{255}x \tag{1}$$

while rounding the pixels to integers μ_i. This was adopted in order to prevent the subsequent noisification to "spill" out of the required dynamic range $[0, 255]$.

Step 2: Computing pixel variance. In Setup I, the variances σ_i^2 were computed using a heteroscedastic model for the photonic (shot) noise for ISO 200 [31], which we parametrized with a scaling factor $C_{\text{ISO}} \in \{0.1, 0.25, 0.5, 1, 2\}$

$$\sigma_i^2 = C_{\text{ISO}} a \mu_i + b, \tag{2}$$

where $a = 6/255$ and $b = 2$. This choice of C_{ISO} maintains an average noise power (MSE) across images in the range $[0, 6]$.

For Setup II, the pixel variances σ_i^2 were computed from the original images from $\mathcal{B}$ using MiPOD's variance estimator [31]. This estimator was originally crafted to capture both the content and noise complexity for the purpose of steganography. By using MiPOD noise model for noisification in Setup II, we are essentially reintroducing the same level of complexity (in terms of MSE) into the denoised image but in a purely stochastic form. The estimated variance σ_i^2 was multiplied by $C_{\text{Mi}} \in \{0.1, 0.15, 0.25, 0.5, 0.8\}$

$$\sigma_i^2 \to \max\{0.01, C_{\text{Mi}} \sigma_i^2\}, \tag{3}$$

to force the mean power of the added noise again to the range $[0, 6]$.

To ensure that all pixels $\mathcal{N}(\mu_i, \sigma_i^2)$ fall inside the 8-bit dynamic range $[0, 255]$ with high probability, the standard deviations σ_i (Eqs. (2) and (3)) were adjusted so that the probability of each pixel falling outside of the required dynamic range was equivalent to a one-sided 5σ Gaussian outlier (2.87×10^{-7}):

$$\sigma_i \to \min\Big\{\frac{1}{5}\min\{\mu_i, 255 - \mu_i\}, \sigma_i\Big\} \triangleq \underline{\sigma}_i. \tag{4}$$

Notice that the noise models in Setup I and II are fundamentally very different. While the ISO noise is only adaptive to luminance, and thus only very weakly adaptive to content, the MiPOD noise strongly depends on the content of the original image and also has a much larger dynamic range. This is why the multiplicative coefficient C_{Mi} is generally smaller than C_{ISO}.

Step 3: Noisifying. To obtain the cover pixel c_i, a sample ξ_i from the normal distribution $\mathcal{N}(0, \underline{\sigma}_i^2)$ was added to μ_i, then rounded to the nearest integer and clipped to the range $[1, 254]$. This ensures that the embedding process can modify all pixels by ± 1 with the same dynamic range $[0, 255]$ for the stego image. Using the square brackets for integer rounding,

[3] For simplicity, we index pixels with a single index $i \in \{1, \ldots, 256^2\}$.

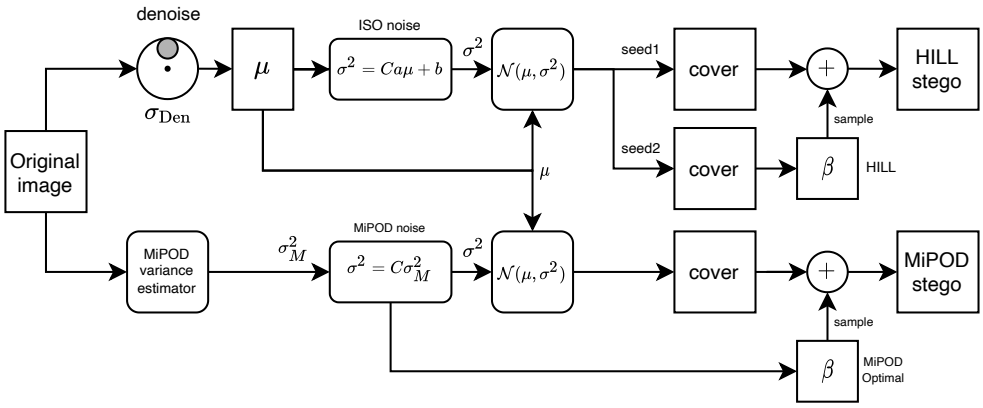


Figure 1: Dataset generation flowchart explaining both setups investigated in this paper. In Setup I (upper branch), heteroscedastic noise is added to denoised images with HILL as the embedding algorithm. In Setup II (lower branch), MiPOD noise is added with embedding optimal to the added noise (MiPOD). The denoising as well as the added noise are parametrized in order to control the content complexity and noise level in each artificial dataset.

$$c_i = [\mu_i + \xi_i]$$
$$c_i = \begin{cases} c_i & \text{if } 1 \le c_i \le 254 \\ 1 & \text{if } c_i \le 0 \\ 254 & \text{if } c_i \ge 255. \end{cases} \tag{5}$$

The i-th cover image pixel thus follows a probability mass function (p.m.f.) $p^{(i)}$ on $\{0, \dots, 255\}$, $c_i \sim p^{(i)}$:

$$p_m^{(i)} = \begin{cases} 0 & m = 0 \\ Q_i\left(m - \frac{1}{2}\right) & m = 254 \\ Q_i\left(m - \frac{1}{2}\right) - Q_i\left(m + \frac{1}{2}\right) & 1 < m < 254 \\ 1 - Q_i\left(m + \frac{1}{2}\right) & m = 1 \\ 0 & m = 255 \end{cases} \tag{6}$$

with $Q_i(x)$ defined as the tail probability of $\mathcal{N}(\mu_i, \sigma_i^2)$:

$$Q_i(x) \triangleq \mathbb{P}\{\mathcal{N}(\mu_i, \sigma_i^2) > x\}. \tag{7}$$

Since we have 5 variances σ^2_{Den} for the denoising filter and five settings for the multiplicative factors (C_{ISO} and C_{Mi}) controlling the power of the imposed noise, there will be $5 \times 5 = 25$ datasets of artificial images with varying degrees of content complexity and added noise.

Figure 2 shows the image '1013.pgm' from BOSSbase from three datasets with the ISO noise in the left column and MiPOD noise in the right column to give the reader a sense of how the artificial images look. The settings for the three datasets correspond (top to bottom) to images with the lowest denoising and lowest noisification, which most closely match the images from $\mathcal{B}$, the strongest denoising and the lowest noisification dataset, which corresponds to the easiest case for steganalysis, and the strongest denoising and strongest noisification dataset formed by images whose content complexity was "replaced" with stochastic complexity (in terms of the MSE).

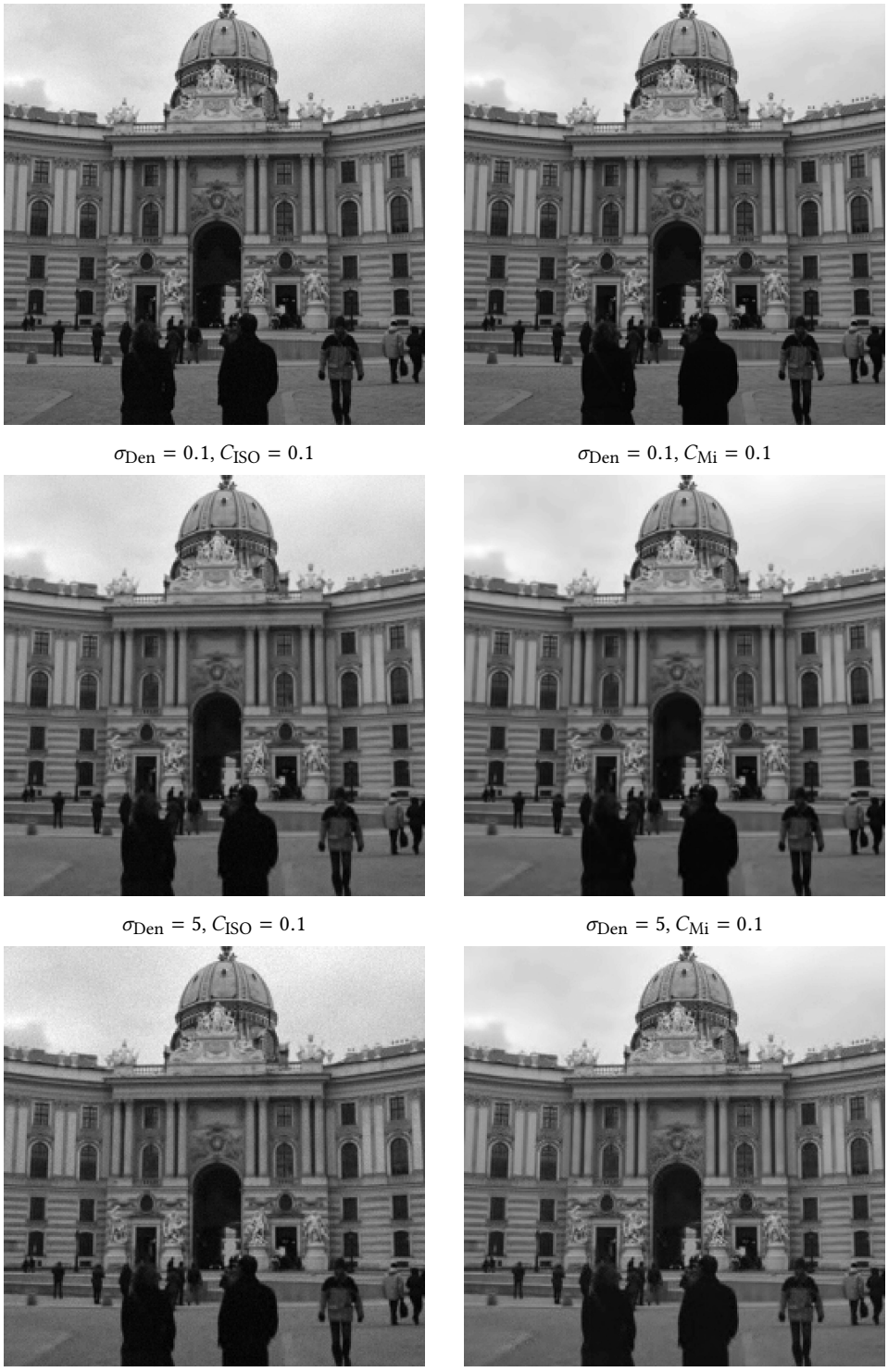
2.2 Embedding algorithms

Both stego algorithms used in this work were implemented with an embedding simulator operating on the rate–distortion bound. The stego signal, $\mathbf{s} = (s_i)_{i=1}^{256^2}$, is a sequence of independent samples from ternary random variables attaining values in $\{-1, 0, +1\}$ with probabilities $\beta_i, 1 - 2\beta_i, \beta_i$ determined by an embedding simulator for each image and payload, which was fixed to 0.4 bits per pixel for both stego methods. Due to curbing the cover pixels to $[1, 254]$, all pixels can be changed by ± 1 with the same probability of both changes. The stego pixel probability mass function (p.m.f.) is a mixture of quantized Gaussians $q^{(i)}$ for all pixels i, $s_i \sim q^{(i)}$,

$$q_m^{(i)} = \begin{cases} \beta_i p_{254}^{(i)} & m = 255 \\ (1 - 2\beta_i) p_m^{(i)} & \\ +\beta_i p_{m-1}^{(i)} & \\ +\beta_i p_{m+1}^{(i)} & 1 \le m \le 254 \\ \beta_i p_1^{(i)} & m = 0. \end{cases} \tag{8}$$

In Setup I, we use the cost-based heuristic algorithm HILL. To avoid dependent stego pixels, we do not compute β_i from the cover image, as would be normally done, but from from another independent noisification (see the upper right section in Figure 1). As will be seen in Section 2.3, this simplification guarantees that stego pixels will be independent, which will simplify the asymptotic form of the most powerful detector.

In Setup II, the embedding is carried out with model-based MiPOD (lower right part of the flowchart) with variances (Fisher information) computed from the original image. Thus, this embedding is optimal in terms of inducing the smallest deflection within the class of all ternary steganographic methods. We note that selecting optimal MiPOD rather than its heuristic version (which estimates the variances from another noisification as in the case of the ISO noise) is crucial for Setup II. This is because the scaled MiPOD variances (3) could be very small, and when estimated only approximately, the embedding would be very detectable at some



$\sigma_{\text{Den}} = 0.1, C_{\text{ISO}} = 0.1$ $\sigma_{\text{Den}} = 0.1, C_{\text{Mi}} = 0.1$

$\sigma_{\text{Den}} = 5, C_{\text{ISO}} = 0.1$ $\sigma_{\text{Den}} = 5, C_{\text{Mi}} = 0.1$

$\sigma_{\text{Den}} = 5, C_{\text{ISO}} = 2$ $\sigma_{\text{Den}} = 5, C_{\text{Mi}} = 0.8$

Figure 2: Image '1013.pgm' from BOSSbase from three artificial datasets corresponding (by rows, top to bottom) to the lowest denoising + lowest noisification, strongest denoising + lowest noisification, and strongest denoising + strongest noisification for the ISO noise (left column) and MiPOD noise (right column). See text for more details.

pixels. Note that this is not as serious of an issue for the ISO noise (Setup I), since the variances (2) are always larger than 2.

2.3 Optimal detectors

Given one specific image $\mathbf{s}$ with pixels s_i, the steganalyst is facing the following statistical hypothesis test for all i :

$$\begin{aligned} \mathcal{H}_0 &: s_i \sim p^{(i)} \\ \mathcal{H}_1 &: s_i \sim q^{(i)}. \end{aligned} \tag{9}$$

For this test, we will assume that the parameters of the added MVG noise, the mean μ_i, and the variance $\underline{\sigma}_i^2$, as well as the change rates β_i are known. Under these assumptions, the test is simple, and, by the statistical independence of pixels in both cover and stego images, the most powerful detector is the log-likelihood ratio

$$\Lambda(\mathbf{s}) = \sum_i \Lambda_{s_i}^{(i)} = \sum_i \log\left(\frac{q_{s_i}^{(i)}}{p_{s_i}^{(i)}}\right), \tag{10}$$

where $\Lambda_m^{(i)} = q_m^{(i)}/p_m^{(i)}$, $m \in \{0, \dots, 255\}$. For convenience, we will use the following normalized form of the log-LRT :

$$\Lambda^\star(\mathbf{s}) = \frac{\sum_i \Lambda_{s_i}^{(i)} - \mathbb{E}_{\mathcal{H}_0}[\Lambda^{(i)}]}{\sqrt{\sum_i \mathrm{Var}_{\mathcal{H}_0}[\Lambda^{(i)}]}}, \tag{11}$$

where

$$\mathbb{E}_{\mathcal{H}_0}[\Lambda^{(i)}] = \sum_m p_m^{(i)} \Lambda_m^{(i)} \tag{12}$$

$$\mathrm{Var}_{\mathcal{H}_0}[\Lambda^{(i)}] = \sum_m p_m^{(i)} (\Lambda_m^{(i)})^2 - \left(\mathbb{E}_{\mathcal{H}_0}[\Lambda^{(i)}]\right)^2. \tag{13}$$

Under the fine quantization limit, $1 \le \sigma_i$ for all i, and as the number of pixels approaches infinity, the Lindeberg's version of the Central Limit Theorem implies

$$\Lambda^\star(\mathbf{s}) \rightsquigarrow \begin{cases} \mathcal{N}(0,1) & \text{under } \mathcal{H}_0 \\ \mathcal{N}(\varrho,1) & \text{under } \mathcal{H}_1 \end{cases}, \tag{14}$$

where $\rightsquigarrow$ means convergence in distribution and $\varrho^2 = 2\sum_i \underline{\sigma}_i^{-4}\beta_i^2 > 0$ is the deflection coefficient. We have verified experimentally for both noise sources and all five parameters by numerically computing the LRT (10) that the fine quantization approximation is tight in terms of the resulting ROCs of the detectors.

The ROC of the most powerful detector *for one specific scene* is thus

$$P_{\mathrm{D}}(P_{\mathrm{FA}}) = Q(Q^{-1}(P_{\mathrm{FA}}) - \varrho). \tag{15}$$

To obtain the ROC for an entire source of scene (dataset), the ROCs for individual scenes (15) should simply be averaged. This kind of ROC is typically highly non-symmetrical and it informs us about the expected P_{D} across the source for a fixed value of P_{FA}. To obtain an equivalent ROC for the ad hoc detector, we would need to train a network for each scene,[4] which would however be computationally infeasible. In practice, ad hoc detectors are trained on just one realization of each scene for many scenes and the ROC is drawn from soft outputs of the trained detector on cover and stego images from the test set. This, however, corresponds to a very different hypothesis testing setup. In particular, the resulting ROC shows the expected P_{D} for an *expected* value of P_{FA}, both expectations taken over the cover source. To properly compare such an ROC with the most powerful detector above, we need to draw the ROC of the *unnormalized* LRT (10) as shown in [16]. Since the non-normalized LRT (10) approximately follows a Gaussian distribution, $\Lambda(\mathbf{s}) \,\dot\sim\, \mathcal{N}(-\varrho/2, \varrho)$ under $\mathcal{H}_0$ and $\Lambda(\mathbf{s}) \,\dot\sim\, \mathcal{N}(\varrho/2, \varrho)$ under $\mathcal{H}_1$ (as also shown in [16], Section 4), given N cover and N stego images from the test set, the ROC of the most powerful detector is

$$P_{\mathrm{D}}(\gamma) = \frac{1}{N}\sum_{k=1}^{N} \mathbb{P}\left(\mathcal{N}(\varrho_k/2, \varrho_k) > \gamma\right) \tag{16}$$

$$P_{\mathrm{FA}}(\gamma) = \frac{1}{N}\sum_{k=1}^{N} \mathbb{P}\left(\mathcal{N}(-\varrho_k/2, \varrho_k) > \gamma\right), \tag{17}$$

where ϱ_k is the deflection coefficient for the k-th scene.

2.4 Heuristic detectors

Since we plan to execute experiments on 25 artificial datasets, the same number of CNN detectors need to be trained. To speed up the training and for the best performance, we used pretrained models as well as seeding. We selected the SRNet [5] and initially trained it for five sources with the smallest noise energy (the lowest C_{ISO} and C_{Mi}) and all five denoising variances σ^2_{Den}. For these five cases, the SRNet was seeded with SRNet pretrained on J-UNIWARD embedded ImageNet (the so-called JIN-SRNet [7]). The network batch size was set to 64 and the training continued till no further improvement was observed on the validation set. For the remaining datasets with stronger noise, we seed the networks with weights corresponding to the same denoising variance and the lowest noise energy while keeping the remaining parameters unmodified. This seeding policy was adopted since the weights from JIN-SRNet were not trained on a dataset statistically close to our artificial datasets.

The network performance was always evaluated on the test set. The performance loss of the networks w.r.t. to the LRT was quantified by the difference between the area under the ROC curve (AUC) of the LRT and the CNN:

$$\Delta = \mathrm{AUC}_{\mathrm{LRT}} - \mathrm{AUC}_{\mathrm{CNN}}. \tag{18}$$

We remind that the AUCs are computed from the ROC for the non-normalized LRT (Eqs. 16 and 17) and from the ROC obtained from the soft output of the CNN detector.

3 EXPERIMENTS

In this section, we compare the performance of the CNNs from Setup I and II to LRT in terms of a loss in AUC (18) evaluated on the test set. The results are interpreted in several different ways to obtain additional insight. In particular, we split the test set into the set of smooth and textured images to see how the performance gap is affected by content complexity. Finally, we study whether the performance gap between the data driven detectors and the optimal LRT could be recuperated by training a CNN for a specific scene, essentially thus allowing the network to learn the scene statistical model.

[4] Since we work with an artificial dataset, we could generate many examples of cover images.

3.1 Data-driven detectors limits

Figure 3 shows the performance loss Δ (18) for both setups as a function of the denoising strength and the power of the added noise. The axes for both of these quantities are expressed in terms of the MSE averaged across the training set for easier interpretability. A larger MSE for denoising means stronger denoising (larger σ_{Den}) and less complex content. The MSE for noisification is the average noise power per pixel per test set image. We wish to point out that for both setups and all 25 datasets, the LRT's performance is nearly perfect, hence Δ mainly accounts for the network loss. Hence, when the network is a random guesser, $\Delta \approx 0.5$, the largest performance loss possible.

For easier interpretation, we mark the four corners shown in the figure with letters: **O** stands for the dataset of images closest to the original images from $\mathcal{B}$ (they only have been slightly denoised with the smallest amount of noise added), **S** marks a dataset with the smoothest content and the smallest amount of added noise (the easiest case for steganalysis). The remaining two corners, **SN** and **ON** (the hardest case for steganalysis) correspond to the same denoising as **S** and **O** but with the strongest added noise.

In Setup I with ISO noise and HILL for embedding (Figure 3 left), by comparing the performance loss in datasets corresponding to the above four corners we observe that adding independent noise does not affect performance as drastically as changing content complexity via denoising. The denoising affects the network performance to a much larger degree especially in the range $0.1 \leq \sigma_{\text{Den}} \leq 1$, which we call the "lip." The lip informs us that high-frequency textures and noise in images from $\mathcal{B}$ significantly contribute to the suboptimality of the CNNs. Once suppressed by denoising ($\sigma_{\text{Den}} > 1$), Δ changes much less w.r.t. denoising and the added noise. Moreover, content complexity negatively affects the CNN more than noise of the same energy (MSE) ($\Delta(\mathbf{O}) \geq \Delta(\mathbf{SN})$).

In Setup II, the data driven detectors experience a more noticeable loss w.r.t. the added noise (compare the increase in Δ between **S** and **SN** and **O** and **ON**). This is because the embedding (MiPOD) is optimal w.r.t. the added noise. Hence, in this setup content complexity affects the CNN performance less negatively than noise of the same energy (MSE) ($\Delta(\mathbf{O}) < \Delta(\mathbf{SN})$). The graph also exhibits the "lip" observed for Setup I.

3.2 Source dissection

In the previous subsection, we analyzed the loss of performance for both setups and commented on the results. The common pattern observed for both setups is the "lip." In order to better understand its onset, we split each dataset into two subsets depending on a heuristic measure of content complexity (as measured in the original images from $\mathcal{B}$) defined as follows. First, we apply the discrete cosine transform (DCT) to disjoint 8×8 blocks in the image and then compute the L_2 norm of DCT coefficients in the 16 highest spatial frequencies (DCT modes $k, l \in \{0, 1, \ldots, 7\}$ with $4 \leq k, l \leq 7$).

Using this content complexity metric, we split each test dataset in two subsets of equal size: smooth and textured images. Using the same CNNs as in the above experiment, we evaluate the performance loss on each subset separately in Figure 4. For both setups, the lip is very pronounced on smooth images but is comparatively much smaller on textured images. This is because the denoising filter acts differently on smooth and textured images. To understand why, we point out that the denoising filter [29] is a Wiener filter in the wavelet domain. To remove additive white Gaussian noise with zero mean and variance σ^2_{Den}, the denoised image is a convex combination of a local average of wavelet coefficients $\hat{v}_i$ and the original coefficient w_i:

$$\hat{w}_i = \hat{v}_i + \frac{\hat{\sigma_i}^2 - \sigma^2_{\text{Den}}}{\hat{\sigma_i}^2}(w_i - \hat{v}_i), \tag{19}$$

where $\hat{\sigma_i}^2$ is a local variance of wavelet coefficients. In textured areas, $\hat{\sigma_i}^2 \gg \sigma^2_{\text{Den}}$ and thus $\hat{w}_i \approx w_i$, which means that the denoising filter is rather conservative, not affecting the image much. In smooth areas, both variances are more likely to become comparable, leading to $\hat{w}_i = \hat{v}_i$, which means the denoising will suppress the noise. Hence, in images that are predominantly smooth, the effect of denoising will be felt sooner than in images with a lot of textured content. This is what creates the lip in the dataset of smooth images and suppresses the lip in textured images.

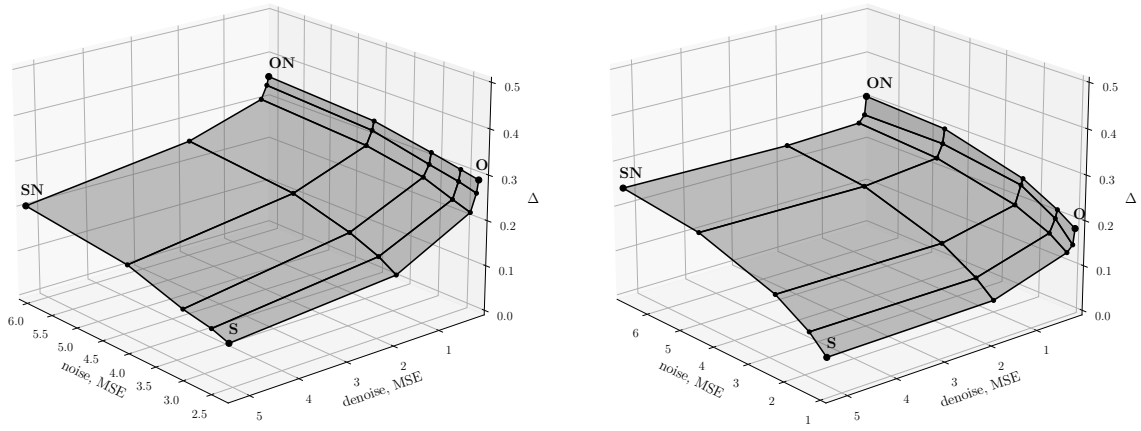
Comparing the left and right columns in Figure 4, it is also clear that the loss of the network detectors w.r.t. the LRT is overall much smaller for smooth images than for textured images.

3.3 Single scene detector

The previous sections revealed the limitations of data driven detectors w.r.t. the content complexity and added independent noise. The loss of performance in terms of the difference between the AUCs of the network detector and the LRT ranges from being quite small (for smooth images) up to 0.3–0.4 for very textured images. Not surprisingly, the LRT has a substantial advantage w.r.t. the network as the parameters of the statistical process generating the cover images are completely known. On the other hand, the network needs to find a rule that *estimates* them by learning this rule from examples of cover and stego images from the training set. This is no easy task especially when the content is complex.

Thus, we next look at a simpler situation for the network by restricting (narrowing) the cover source to a single scene (also called acquisition oracle). In simple words, we allow the network to learn detecting steganographic changes in different realizations of a fixed cover image. Mathematically, we generate multiple arrays of 256×256 Gaussian variables $\mathcal{N}(\mu_i, \sigma_i^2)$ (here, μ_i is the noise free-scene). To this end, we experimented with two scenes from the dataset, one with smooth (BOSSbase image '9388.pgm') and one with textured content (image '680.pgm') w.r.t. the content complexity metric explained above. We generate the training, validation, and testing sets of the same sizes as for the datasets above. This time, all images are, however, different realizations of a fixed scene. We train the same network (SRNet) pretrained with JIN-SRNet until the validation loss starts increasing to prevent overfitting. For this experiment, we selected only three versions of these datasets corresponding to the **O**, **S**, and **SN** corners.

Figure 5 shows that for the smooth scene, both setups, and all three types of the datasets **O**, **S**, and **SN**, the SRNet trained on the acquisition oracle of that scene achieves almost the same performance (AUC) as the LRT. While there is still some performance loss for the textured scene, for the dataset closest to the original image (**O**), this loss is very small.



Setup I (ISO noise, HILL) Setup II (MiPOD noise, MiPOD)

Figure 3: Performance loss Δ (18) between AUCs of ROCs drawn for the most powerful detector (non-normalized LRT) and a data driven CNN as a function of denoising strength and noise power (in terms of MSE). The four labeled corners correspond to smooth (S), closest to original (O), smooth and fully noisified (SN), and original and fully noisified (ON).

4 CONCLUSIONS

How much can they still improve? This question is as old as the first machines trained to detect steganography yet it is still left largely unanswered. In this paper, we attempt to shed some light on this difficult but relevant problem. Due to the complexity of digital images, there is little hope that tractable optimal detectors will ever be built for real images. Hence, we form a family of sources with a known statistical model so that it is possible to detect steganography optimally with a likelihood ratio test. The creation of these sources was parametrized in order to obtain control over the content complexity (via a denoising filter) and the amount of independent but not white additive Gaussian noise. The goal was to learn how both contribute to the limits of modern machine learning detectors (SRNets). To substantiate our conclusions, we investigated two different setups – noisification with a heteroscedastic sensor noise and with noise determined by MiPOD's variance estimator. Measuring the accuracy loss in terms of the difference between AUCs of the CNN detector and the corresponding LRT, we learned that

- SRNet's loss depends both on content complexity left behind after denoising as well as the power of the added noise.
- This loss is generally much smaller for smooth images than for textured images.
- For suboptimal (heuristic) steganography, the noise is less damaging than content (when both are measured with MSE).
- For steganography optimal to the added noise, this conclusion is the opposite.
- A network trained on many acquisitions of a fixed scene will recuperate most of the loss, suggesting that networks have the capacity to approximately learn the parameters of a cover source narrowed to a fixed scene.

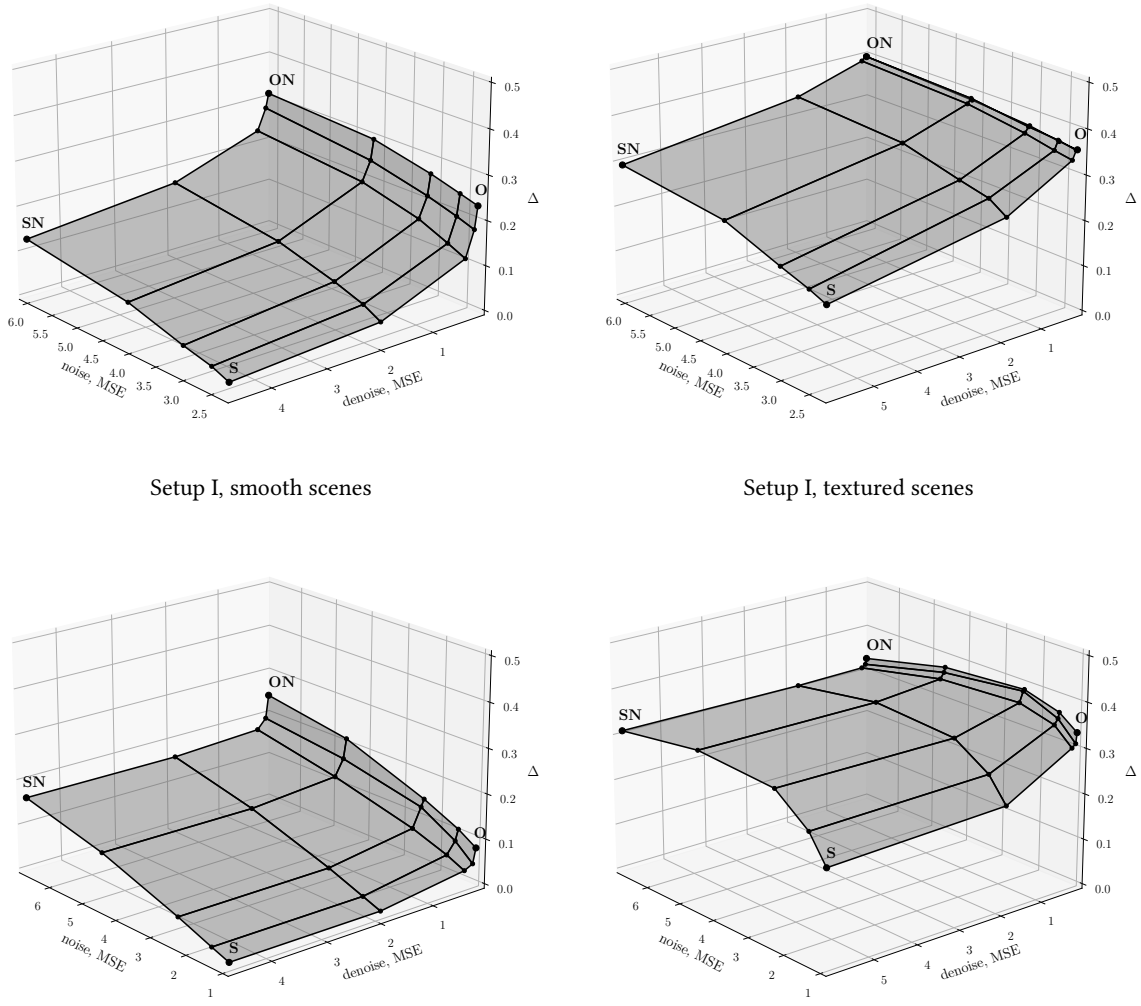
Our study has many limitations. For starters, we do not work with realistic models of digital images. A better and much more complex approach would be to work with models in the RAW, undeveloped domain, and approximate the developed domain with a stochastic lattice (MVG) suitably simplified (such as in [37]) to permit evaluating security with a most powerful detector. The experiments were carried out on variants of BOSSbase, which is a very complex cover source due to the rather aggressive downsampling from the original RAW size images. Perhaps, for cover sources with higher resolution, the local content will be significantly smoother, giving a better chance to data driven detectors to operate closer to optimal. Finally, we limited our study to the spatial domain leaving the JPEG domain as part of our future effort.

5 ACKNOWLEDGEMENTS

The work on this paper was supported by NSF grant No. 2028119.

REFERENCES

[1] P. Bas. An embedding mechanism for natural steganography after down-sampling. In *IEEE ICASSP*, New Orleans, LA, March 5-9, 2017.
[2] 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.
[3] S. Bernard, P. Bas, T. Pevný, and J. Klein. Optimizing additive approximations of non-additive distortion functions. In D. Borghys and P. Bas, editors, *The 9th ACM Workshop on Information Hiding and Multimedia Security*, pages 105–112, Brussels, Belgium, June 22–25, 2021.
[4] R. Böhme. *Improved Statistical Steganalysis Using Models of Heterogeneous Cover Signals*. PhD thesis, Faculty of Computer Science, Technische Universität Dresden, Germany, 2008.
[5] 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.
[6] M. Boroumand, J. Fridrich, and R. Cogranne. Are we there yet? In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking,*



Setup I, smooth scenes

Setup I, textured scenes

Setup II, smooth scenes

Setup II, textured scenes

Figure 4: Performance loss Δ (18) between AUCs of ROCs drawn for the most powerful detector (non-normalized LRT) and a data driven CNN as a function of the denoising strength and noise power (in terms of MSE) on smooth (left column) and textured scenes (right column).

Security, and Forensics 2019, San Francisco, CA, January 14–17, 2019.

[7] J. Butora, Y. Yousfi, and J. Fridrich. How to pretrain for steganalysis. In D. Borghys and P. Bas, editors, *The 9th ACM Workshop on Information Hiding and Multimedia Security*, Brussels, Belgium, 2021. ACM Press.

[8] K. Chubachi. An ensemble model using CNNs on different domains for ALASKA2 image steganalysis. In *IEEE International Workshop on Information Forensics and Security*, New York, NY, December 6–11, 2020.

[9] R. Cogranne and J. Fridrich. Modeling and extending the ensemble classifier for steganalysis of digital images using hypothesis testing theory. *IEEE Transactions on Information Forensics and Security*, 10(2):2627–2642, December 2015.

[10] 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.

[11] R. Cogranne, Q. Giboulot, and P. Bas. ALASKA–2: Challenging academic research on steganalysis with realistic images. In *IEEE International Workshop on Information Forensics and Security*, New York, NY, December 6–11, 2020.

[12] R. Cogranne, V. Sedighi, T. Pevný, and J. Fridrich. Is ensemble classifier needed for steganalysis in high-dimensional feature spaces? In *IEEE International Workshop on Information Forensics and Security*, Rome, Italy, November 16–19, 2015.

[13] T. Denemark, P. Bas, and J. Fridrich. Natural steganography in JPEG compressed images. 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.

[14] T. Denemark, M. Boroumand, and J. Fridrich. Steganalysis features for content-adaptive JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 11(8):1736–1746, August 2016.

[15] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich. Selection-channel-aware rich model for steganalysis of digital images. In *IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, December 3–5, 2014.

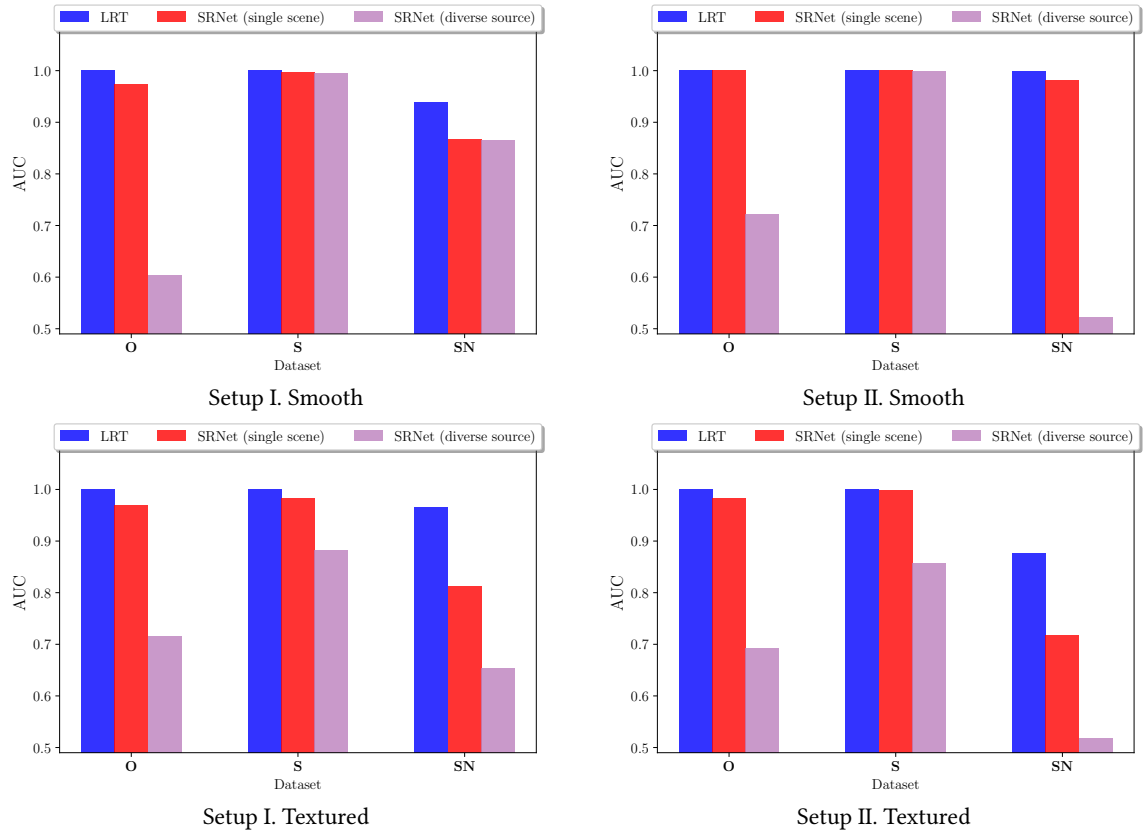


Figure 5: AUC of the LRT (blue), of the SRNet trained on the diverse training dataset as in Section 2 (purple), and of SRNet trained on multiple realizations of the same fixed scene (red). AUC is computed on the test set consisting of different realizations of the same scene. The textured scene is scene '680.pgm', while the smooth scene is '9388.pgm'.

[16] E. Dworetzky, E. Kaziakhmedov, and J. Fridrich. On comparing ad hoc detectors with statistical hypothesis tests. In Y. Yousfi, A. Bharati, and C. Pasquini, editors, *The 11th ACM Workshop on Information Hiding and Multimedia Security*, Chicago, IL, June 28–30, 2023. ACM Press.

[17] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, Oct. 2008.

[18] 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.

[19] G. E. Healey and R. Kondepudy. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, March 1994.

[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] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li. A novel image steganography method via deep convolutional generative adversarial networks. *IEEE Access*, 6:38303–38314, 2018.

[22] J. Huang, J. Ni, L. Wan, and J. Yan. A customized convolutional neural network with low model complexity for JPEG steganalysis. 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.

[23] J. R. Janesick. *Scientific Charge-Coupled Devices*, volume Monograph PM83. Washington, DC: SPIE Press - The International Society for Optical Engineering, January 2001.

[24] 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.

[25] B. Li, M. Wang, and J. Huang. A new cost function for spatial image steganography. In *Proceedings IEEE, International Conference on Image Processing, ICIP*, Paris, France, October 27–30, 2014.

[26] 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.

[27] L. Li, W. Zhang, C. Qin, K. Chen, W. Zhou, and N. Yu. Adversarial batch image steganography against CNN-based pooled steganalysis. *Signal Processing*, 181:107920–107920, 2021.

[28] Ce Liu, R. Szeliski, Sing Bing Kang, C.L. Zitnick, and W.T. Freeman. Automatic estimation and removal of noise from a single image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):299–314, Feb 2008.

[29] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters*, 6(12):300–303, December 1999.

[30] 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.

[31] V. Sedighi, R. Cogranne, and J. Fridrich. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 11(2):221–234, 2016.

[32] Y. Q. Shi, P. Sutthiwan, and L. Chen. Textural features for steganalysis. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 63–77, Berkeley, California, May 15–18, 2012.

[33] 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.

[34] W. Su, J. Ni, X. Hu, and J. Huang. New design paradigm of distortion cost function for efficient JPEG steganography. *Signal Processing*, 190:108319, 2022.

[35] T. Taburet, P. Bas, J. Fridrich, and W. Sawaya. Computing dependencies between DCT coefficients for natural steganography 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.

[36] T. Taburet, P. Bas, W. Sawaya, and J. Fridrich. A natural steganography embedding scheme dedicated to color sensors in the JPEG domain. 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.

[37] T. Taburet, P. Bas, W. Sawaya, and J. Fridrich. Natural steganography in JPEG domain with a linear development pipeline. *IEEE Transactions on Information Forensics and Security*, 16:173–186, 2021.

[38] W. Tang, B. Li, M. Barni, J. Li, and J. Huang. Improving cost learning for JPEG steganography by exploiting JPEG domain knowledge. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):4081–4095, 2021.

[39] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang. CNN-based adversarial embedding for image steganography. *IEEE Transactions on Information Forensics and Security*, 14(8):2074–2087, 2019.

[40] W. Tang, H. Li, W. Luo, and J. Huang. Adaptive steganalysis based on embedding probabilities of pixels. *IEEE Transactions on Information Forensics and Security*, 11(4):734–745, April 2016.

[41] Thanh Hai Thai, R. Cogranne, and F. Retraint. Camera model identification based on the heteroscedastic noise model. *Image Processing, IEEE Transactions on*, 23(1):250–263, Jan 2014.

[42] 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.

[43] J. Yang, K. Liu, X. Kang, E. K. Wong, and Y.-Q. Shi. Spatial image steganography based on generative adversarial network, 2018.

[44] J. Yang, D. Ruan, J. Huang, X. Kang, and Y. Q. Shi. An embedding cost learning framework using GAN. *IEEE Transactions on Information Forensics and Security*, 15(10):839–851, 2020.

[45] J. Yang, D. Ruan, X. Kang, and Y.-Q. Shi. Towards automatic embedding cost learning for JPEG steganography. 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.

[46] J. Yang, Y.-Q. Shi, E.K. Wong, and X. Kang. JPEG steganalysis based on densenet. *CoRR*, abs/1711.09335, 2017.

[47] 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.

[48] 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.

[49] Y. Yousfi, J. Butora, and J. Fridrich. CNN steganalyzers leverage local embedding artifacts. In *IEEE International Workshop on Information Forensics and Security*, Montpellier, France, December 7–10, 2021.

[50] 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.

[51] Y. Yousfi, J. Butora, E. Khvedchenya, and J. Fridrich. ImageNet pre-trained CNNs for JPEG steganalysis. In *IEEE International Workshop on Information Forensics and Security*, New York, NY, December 6–11, 2020.

[52] Y. Yousfi and J. Fridrich. An intriguing struggle of CNNs in JPEG steganalysis and the OneHot solution. *IEEE Signal Processing Letters*, 27:830–834, 2020.

[53] 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.

[54] J. Zeng, S. Tan, G. Liu, Bin Li, and J. Huang. WISERNet: Wider separate-then-reunion network for steganalysis of color images. *CoRR*, abs/1803.04805, 2018.

[55] N. Zhong, Z. Qian, Z. Wang, X. Zhang, and X. Li. Batch steganography via generative network. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(1):88–97, January 2021.