Size-Independent Reliable CNN for RJCA Steganalysis

Jan Butora and Patrick Bas, *Senior Member, IEEE*

***Abstract*—Detection of image steganography is principally implemented with supervised machine learning detectors. There are two main drawbacks to this approach: the detectors are overly specific to a given image source, and the performance guarantees are only empirical. In this work, we further study a previously proposed deep learning detector that exploits natural image structure imposed by JPEG compression with high quality. We show in a controlled environment that for a fixed JPEG compressor, the soft outputs of a deep learning classifier - the logits - follow a Gaussian distribution. We prove a scaling law stating that the variance of this distribution scales linearly with the image size. By disabling padding in the convolutional neural network, we demonstrate that the mean of the logit distribution does not change, allowing us to directly analyze images of different sizes. Focusing on the logits, we show that we can prescribe a threshold with a theoretical false positive rate for a wide range of image sizes, which is then closely satisfied on real cover images, even for small probabilities such as 10^{-4}. Moreover, the detection power on steganographic images still generalizes to non-adaptive and content-adaptive steganography.**

***Index Terms*—Steganalysis, false positive control, JPEG, arbitrary size**

I. Introduction

A lot of emphasis on privacy and security has been used in today's communication. Private communication tools have been thus increasingly more important in society. Steganography is one of these tools that allows the modification of a given digital image (cover image) to communicate a desired secret message. The only condition is that the modifications are statistically untedetectable [22]. Two main branches of image steganography have existed since the early beginning of the field: spatial steganography using the pixel representation of the image, and JPEG steganography using the DCT coefficients of the compressed image.

On the other hand, steganalysis aims to detect the mere presence of steganography. Many targeted attacks against early steganography have been proposed [3], [18], [19], [41], [42]. Unfortunately, with the development of content-adaptive steganography [6], [14], [27], [37], [40], the models targeted in the previous attacks no longer hold and are thus unusable. Researchers have hence started using machine learning classification approaches by collecting steganography-specific features [23], [28], [34]. More recently, deep learning (DL) detectors have been shown to be the most accurate detectors available [10], [12], [47]–[49]. Despite their superior detection abilities, their performance guarantees are only empirical due to their supervised training. To make things worse, training convolutional neural networks (CNNs) on images of variable sizes is practically unfeasible, which forces the steganalyst to train their detector on images of fixed size. This is a major drawback of this DL strategy since in practice, we can encounter images of any possible size. Additionally, it has been shown that the CNNs suffer from the Cover-Source Mismatch (CSM) [33], [35] more than the feature-based methods [26], [50], which further decreases the quality of the empirical error rates of a given detector.

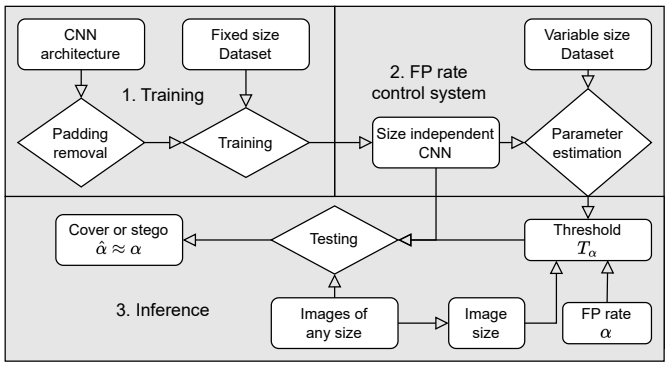

Figure 1: A diagram of the proposed method for size-independent steganalysis. Contrary to classical steganalysis, our system proposes also to control False Positive (FP) rates.

The recent Reverse JPEG Compatibility Attack (RJCA) [8], [9], [11] is a steganalysis method exploiting the structure of the JPEG images. This attack inspects the rounding errors of the decompressed JPEG files and it has been shown that steganography changes first-order statistics of this signal, which can be very accurately detected with a DL detector. Moreover, such a signal is rather insensitive to image processing operations prior to the JPEG compression, making it very robust against the CSM. On the other hand, we showed in our previous work that the attack could suffer from a JPEG compressor mismatch [7].

For the steganalyst to decide if the test image is either Cover or Stego, one minimal requirement is to control the False Positive (FP) rate of the classifier[1], that is why metrics such as MD5 (miss detection rate for a FPR of 5%) were also proposed to benchmark steganographic schemes [12].

[1]Misclassifying cover image as a stego image.

For a binary classifier, this means finding a decision threshold corresponding to a desired FP rate. However, with a DL detector, the only way, so far, is to empirically compute the decision threshold from the data. Another approach, without DL, uses Kolmogorov-Smirnov test [36] on the decompression rounding errors [20], [21] (see Section I-A2 for more details) but the detection power is in this case greatly reduced.

In this work, we take the RJCA further by studying the CNN detector's logits coming from cover images. This allows us to find an accurate model of the cover logit distribution, which provides us with theoretical thresholds for any desired FP rate. We also show how this distribution changes with image size, giving us the ability to steganalyze images of any size with a prescribed FP rate without the need for retraining a CNN detector trained on images of fixed size. Of course, one can always take crops of bigger images to avoid dealing with different image sizes, but this way the detector does not pool all the available information and, as we will see in Section IV it is possible to design a detector which gets more and more accurate with increasing image size.

A. Prior Art

We recall that the two main goals of this work are (1) deep learning steganalysis of arbitrarily sized images and (2) control of the detector's FP rate.

Even though both of these topics have been previously studied, none of the proposed approaches is able to achieve both goals at the same time.

1) Images of Arbitrary Size: While steganalysis of images of any size is straightforward to implement with the feature-based methods (feature vector from any image is of the same dimension), this changed with the development of DL architectures [4], [39], [44], [46]. Due to the mini-batch-driven supervised training of the CNNs, all images in a given mini-batch must be of the same size. Despite this, training the CNNs with differently sized mini-batches is possible, but in practice would be rather cumbersome. And even then, the mini-batches are limited by GPU memory, meaning we cannot use non-trivial mini-batches of images of large sizes. In the field of computer vision, this is mitigated by simply resizing every image on the input to a desired size [17]. However, such an approach cannot be used for steganalysis for two obvious reasons: (1) resizing of the image destroys the signal of interest (the stego signal) and (2) it prevents pooling all the available information [31].

A different, two-step approach has thus been proposed in [24]. The authors first trained a slightly modified YeNet [45] on images of a fixed size (tiles), as typically done in the steganographic community. Moreover, the following change was added to the CNN architecture to better capture the information about the image size. In the majority of cases, the last layers of a CNN detector consist of a Global Average Pooling (GAP) across spatial dimensions, providing a vector of size C,[2] followed by a fully-connected layer, resulting in two logits - one for the cover class and one for the stego class. This is then typically followed by a softmax activation. The authors changed this structure and extracted not only the average, but also variance, minimum, and maximum, creating thus a vector of size $4C$. After having trained this so-called Tile Detector (TD), features of size $4C$ were then extracted from every image of interest (with potentially different sizes) and were used to train a two-layer Multi-Layered Perceptron (MLP). Such MLP then served as a steganalysis detector of arbitrarily sized images. A similar approach was later applied to SRNet [4] in [48] during the first ALASKA competition [12].

2) Control of the False Positive Rate: In a typical deep steganalysis work, a CNN detector is trained to distinguish between cover and stego classes by utilizing the cross-entropy loss function. The detector's performance is then measured as the minimum probability of error under equal priors

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

where P_{FA} is the probability of False Alarm (FP rate), and P_{MD} is the probability of Missed Detection - misclassifying stego image as a cover (false negative rate). This value corresponds to a single point on the detector's ROC curve and does not inform us at all about its behavior for different values of P_{FA}, which is impractical in an operational setting, where a steganalyst wants to set its detector's threshold to a very small value, such as 10^{-3} or 10^{-4}. Of course, one can set up a threshold to achieve such P_{FA} empirically but practically this is unfeasible since we have a limited amount of training samples (usually only tens of thousands of images).

To mitigate this issue, we can see in the literature the use of statistical tests [15] or p-values [30], [43]. In recent works [20], [21], the decompression rounding errors from cover images are split into 64 lattices (one per each pixel position in an 8×8 block) and two-sample Kolmogorov-Smirnov test is performed on each lattice. The p-values of every test are corrected by the Bonferroni procedure [36] to increase the power of the detector by aggregating different observations. In this context, the desired (even very low) P_{FA} can be theoretically found, however, the stego detection power of such a detector is greatly lacking compared to a DL approach.

Note also that controlling the FPR can also be seen as a calibration problem [25] in machine learning. These methods are for example used in steganalysis to fuse comparable outputs of different classifiers [2], [48]. However, calibration techniques are not designed to face very small FPR.

[2] C denotes the number of channels after the last convolutional layer in a given architecture.

B. Our Contribution

The goal of this work is to develop a universal steganography detector for JPEG images compressed at Quality Factor 100 which can be used for analyzing images of arbitrary size, detecting potentially any steganography, and for which we have perfect control of its false positive rate. The main contributions can be summarized by the following:

- We introduce a Gaussian model of a DL detector's soft outputs (logits), allowing us to control the detector's FP rate. We verify, that the theoretical Gaussian threshold for any given FP rate corresponds to an empirical FP rate given by the data.
- We show that by disabling padding in the CNN architecture, the mean of the logit distribution is independent of the image size and consequently, the FP rate relies only on the variance of the distribution.
- We find the *variance scaling law* - an affine relationship between the variance of the Gaussian distribution and the reciprocal image size. We are thus able to perfectly predict the Gaussian distribution for any image size, without the need to retrain the detector on images of other sizes. In other words, we are able to control the FP rate for images of arbitrary size.
- By limiting ourselves only to Quality Factor 100 JPEG images, where we can very accurately model the cover distribution, we are able to build a one-class classifier and thus reliably detect previously unseen steganography in images of any size.

C. Organization of the Paper

The rest of the paper is organized as follows: In the next section, we introduce the basic concepts and notations. Section III introduces the dataset and detectors used throughout the paper. In Section IV we describe and evaluate the proposed methodology. We compare our method to previous state-of-the-art in Section V and the paper is then concluded in Section VI.

II. Preliminaries and Notation

In this Section, we introduce several concepts and notations that will be used throughout the paper.

Boldface symbols are reserved for matrices and vectors. Rounding x respectively to the nearest integer and the nearest higher integer will be denoted $[x]$ and $\lceil x \rceil$. Denote $\otimes$ element-wise multiplication and $\oslash$ element-wise division. Let $\mathrm{DCT}(\cdot)$ denote the 2D type-II Discrete Cosine Transform (DCT) used during the JPEG compression and $\mathrm{IDCT}(\cdot)$ its inverse (the 2D type-III DCT).

To compress a grayscale image using the JPEG format, we take each 8×8 block of uncompressed pixels $\mathbf{x}$ and transform them to DCT coefficients

$$\mathbf{c} = \left[\mathrm{DCT}(\mathbf{x} - 128) \oslash \mathbf{q}\right], \tag{2}$$

where $\mathbf{q}$ is an 8×8 integer-valued quantization matrix defined by the JPEG Quality Factor (QF). To decompress the DCT coefficients back to pixel values, the steps are reversed

$$\mathbf{y}' = \left[\mathrm{IDCT}(\mathbf{c} \otimes \mathbf{q}) + 128\right]. \tag{3}$$

Note that since JPEG is a lossy compression (due to the rounding operation and numerical imprecisions in the DCT), the uncompressed image $\mathbf{x}$ and the decompressed image $\mathbf{y}'$ are not necessarily the same.

A. Reverse JPEG Compatibility Attack

We briefly recall the basic idea of the RJCA [8] as we will use it to build our detector. The main constraint is limiting ourselves only to QF 100. However, as previously noted in [12], it is also a very popular quality factor since 14% of all images uploaded to Flickr have been compressed with this quality. For lower quality factors, the variance of the signal of interest gets too big and the signal becomes useless. While Eq. (3) is what many decompressing libraries would provide a user with, we can avoid the rounding operation, to prevent the extra information loss. We thus decompress the DCT coefficients into non-integer pixel values

$$\mathbf{y} = \mathrm{IDCT}(\mathbf{c}) + 128, \tag{4}$$

where we left out the quantization table, as it is composed of ones at the highest QF. It has been shown in the original publication that the decompression rounding errors, denoted by

$$\mathbf{e} = \mathbf{y}' - \mathbf{y}, \tag{5}$$

when computed on cover images follow a Wrapped Gaussian distribution. It then follows that steganography necessarily increases the variances of this distribution, which can be very reliably detected even for small embedding payloads. The best performance is obtained with a CNN detector, where the rounding errors $\mathbf{e}$ are provided as inputs instead of the decompressed images. For more information about the attack, we refer the reader to [8].

B. Convolutional Neural Networks

To analyze how CNN logits change w.r.t. the image size, it is important to understand some inner mechanisms of such architectures, namely the *receptive field* and *padding* which combined lead to *feature poisoning*. We shall see in Section IV that feature poisoning impacts the mean of the logit distribution and that the receptive field induces a subset of correlated features.

1) Receptive Field: The Receptive Field of a CNN is the region of the input that contributes to the feature after the last convolutional layer. Let k_i and s_i represent respectively the filter size and stride of the i-th convolution and let L be the number of convolutions in a given CNN architecture. It was shown [1] that the receptive field size R at the input image can be computed as

$$R = \sum_{l=1}^{L} \left((k_l - 1) \prod_{i=1}^{l-1} s_i \right) + 1. \tag{6}$$

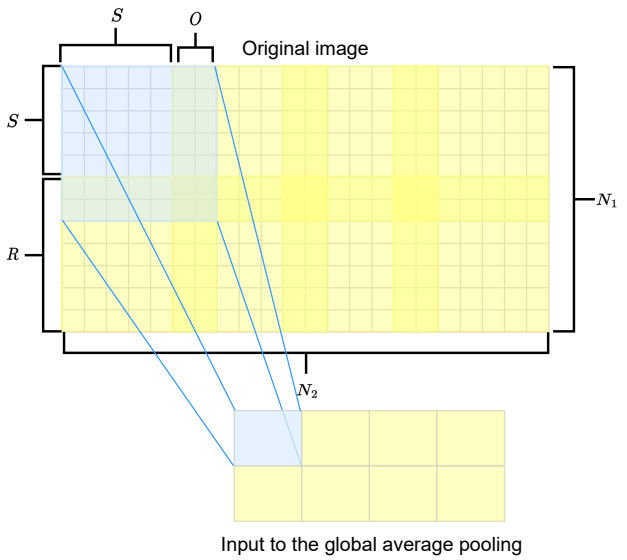

Figure 2: Illustration of overlapping neighboring receptive fields in a convolutional neural network.

The stride of the receptive field can be computed as a product of all strides

$$S = \prod_{l=1}^{L} s_i. \tag{7}$$

The overlap of the neighboring receptive fields can then be computed as

$$O = R - S. \tag{8}$$

Assuming no padding is applied in the CNN, it follows that for an image of size $N_1 \times N_2$, the feature size after the last convolutional layer (or equivalently before the global average pooling) is of size $\lceil (N_1 - R + 1)/S \rceil \times \lceil (N_2 - R + 1)/S \rceil$, see Figure 2 for a graphical representation. As the feature size has to stay positive, this gives a lower bound on the input image size: $N_1, N_2 \geq R$. This constraint can be effectively mitigated by using padding in the network, but as we will see in Section IV, we do not want to use padding for the detector as it changes the distribution of the inputs. In computer vision tasks, changing the distribution with padding is typically not an issue as it can be countered by rather aggressive data augmentations, such as resizing. These augmentations are, however, not applicable in steganalysis because they would destroy the weak stego signal of interest.

In our experiments, we use only the SRNet [4], for which we found $R = 179, S = 16$, and $O = 163$. We want to point out, that without padding, the size of the smallest JPEG image we could analyze with this architecture is 184×184 (the sizes need to be multiples of 8). Note also, that the size of the overlap of the neighboring receptive fields in SRNet is 163×179. For EfficientNet [39] (EFN) B0 and B4, we found $R = 851, S = 32$ and $R = 1799, S = 32$ respectively, which clearly shows why padding has to be used in these architectures since otherwise, the minimum image size would be too big.

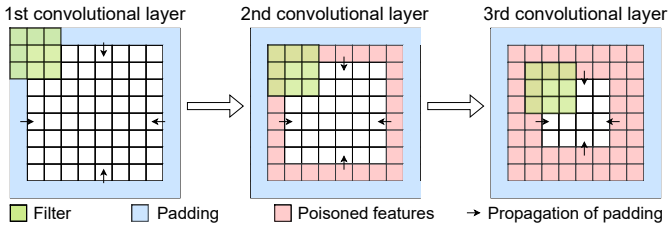

Figure 3: Propagation of the feature poisoning through successive layers due to padding.

2) Feature Poisoning: In this Section, we explain the need to discard padding with the concept of *Feature poisoning*. Many[3] CNN architectures implicitly use some kind of padding during their convolution operations (e.g. zero padding, mirror padding, etc.) in order to preserve the feature dimensions and to avoid having the minimum image size too big as we have just seen. While in computer vision tasks, this might be justifiable, we are questioning the usefulness of padding in steganalysis. We will see in Section IV-A that padding, especially zero padding, is, in fact, undesirable because convolution even with a small 3×3 kernel, creates features with different properties than the rest of the image. We call these features *poisoned*. Figure 3 demonstrates this phenomenon with three convolutional layers, each having a kernel of size 3×3. Note that the stride does not have a serious effect on the feature poisoning because the padding is usually applied only on an outside layer of the image. Before the first layer, there is only the outer padding (depicted in blue) and no poisoned features. The first convolution creates a layer of poisoned features (depicted in red). Consequently, every subsequent convolution will create an extra layer of poisoned features, which is why we can see two layers (two pixels from each boundary) of poisoned features after the second convolutional layer. We will refer to the number of poisoned layers as the poisoned size.

Let us denote the poisoned size on the input to the i-th convolutional layer as P_i. Next, it is rather straightforward to realize that the poisoned size dependent on the kernel stride s_{i-1}, padding size p_{i-1} (which is typically dependent on the kernel size k_{i-1}) and on the previous poisoned size P_{i-1}. We can then find that

$$P_{i+1} = \left\lceil \frac{P_i + p_i}{s_i} \right\rceil, \tag{9}$$

where $P_0 = 0$ is the initial poisoned size.

Unfortunately, modern CNNs, such as EfficientNet [39] or SRNet, contain many convolutions, each using padding, and many of them having stride equal to 1. In Figure 4, we show for several architectures a relative size of poisoned features in the pre-GAP feature - the last feature before the global average pooling. We can see that for image sizes smaller than the receptive field size R, all the features are poisoned. Moreover, as discussed in the previous section,

[3]We are not aware of an architecture that would implicitly not use padding.

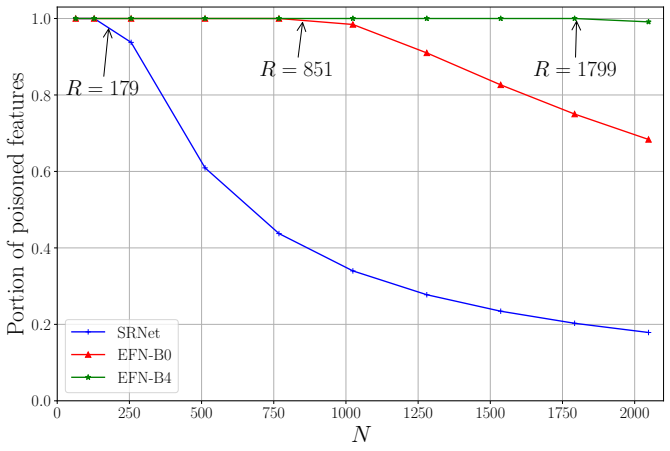

Figure 4: Relative size of poisoned features for images of size $N \times N$ for different architectures with receptive field size R.

we can see that the receptive field size R of the EfficientNet is much larger, essentially forcing the use of padding. Note that this larger value of R is due to larger filters (most of the convolution kernels are of size 5×5, adding two poisoned layers) and due to subsampling early in the network, while SRNet tends to subsample only deeper in the network.

III. Experimental Setup

Before studying the output of the trained detector, we detail how the detector is trained in this section.

A. Training and testing dataset

The dataset used throughout this work is the ALASKA 2 [13] comprised of 80k uncompressed grayscale images of size 2048×2048. Using ImageMagick's `convert`, we first JPEG compress the dataset with QF 100 to create the cover set. We then split the dataset into training, validation, and testing sets of sizes 35k, 5k, and 40k respectively. For the stego images, we embed the cover set by simulating optimally coded ternary Least Significant Bit Matching (LSBM) at payloads $0.3, 0.05$, and 0.01 bits per DCT coefficient (bpc). We chose a non-adaptive steganographic scheme, as it was shown that constant embedding costs provide the best security against the RJCA due to the smaller number of embedding changes [5]. To test the detector on unseen, adaptive steganography, we also embedded the cover sets with UERD [27] and J-UNIWARD [29] at the same payloads. To evaluate our methodology on images of different sizes, we crop the cover set into smaller images of size $N \times N$. We will be using two sets of image size N: $\mathcal{S}_{train} = \{184, 256, 512, 768, 1024, 1280, 1536, 1792, 2048\}$[4] and $\mathcal{S}_{test} = \{1000, 1200, 1400, 1600, 1800, 2000\}$.

[4]The smallest value corresponds to the smallest JPEG image size bigger than SRNet's receptive field size.

Note that to prevent storing a huge amount of data, we simply center crop the original images at resolution 2048×2048 when needed. This applies also to stego images, which should not create discrepancies for the non-adaptive LSBM. For the UERD and J-UNIWARD images, the resulting payload of the cropped images could be potentially different due to its adaptive nature, but we will consider this effect negligible.

B. Detectors

Based on the preliminary analysis in Section II, we choose the SRNet [4], because of its small receptive field size. We train this detector on cover and LSBM images with 0.3 bpc payload, using only their rounding errors $\mathbf{e}$ (see (5)) as inputs. For this reason, we will refer to this detector as the eSRNet. Only crops of size 512×512 are used during the training of this detector. The detector is trained for 10 epochs with the rest of the hyperparameters as explained in [7] (see Section 3.2). Additionally, we disable the padding in every convolutional layer, as will be explained in Section IV-A.

1) Variable Size Detector: We train another eSRNet (also without padding) on images of variable size (VS) in $\mathcal{S}_{train}$. Note that such an approach is not unrealistic, we simply need to provide images of the same dimension in every mini-batch. We do this in practice by taking cover and stego images of size 2048×2048 and simply center-cropping the entire mini-batch into the desired size. Due to increased memory requirements, we used batch norm synchronization over 4 GPUs, each having a mini-batch size of 8. We refer to this detector as VS-eSRNet.

2) Prior Art Detector: To implement the detector from [48], we need to proceed in two steps, as previously explained in Section I-A1. First, an eSRNet tile detector (TD) is trained similarly as described in the previous section on images of size 512×512. However, we modify the SRNet structure to produce a feature vector f of size 4×512 instead of 512 by extracting after the last convolution the average (GAP), but also the variance, minimum, and maximum over each channel. This vector is then fed to the fully connected layer.

As mentioned in [48], the trained TD is then used as a feature extractor. For image crops of every size in $\mathcal{S}_{train}$, we extract their features f. Note that for every image size, f is of size 4×512. Using the same training, validation, and testing split as in the TD training, we train an MLP with 1 hidden layer of size 8×512 on these features. The batch size of the MLP is set to 100 and the other hyperparameters are kept the same. To evaluate the MLP on images of unseen sizes, that is of those in $\mathcal{S}_{test}$, we use TD to extract their features f and then feed these to the trained MLP.

IV. Studying the Logits

In this Section, we describe the proposed methodology for finding the detector's logit (soft output) distribution and using it for the steganalysis of images of any size. Note that all the detectors used in this section are fixed

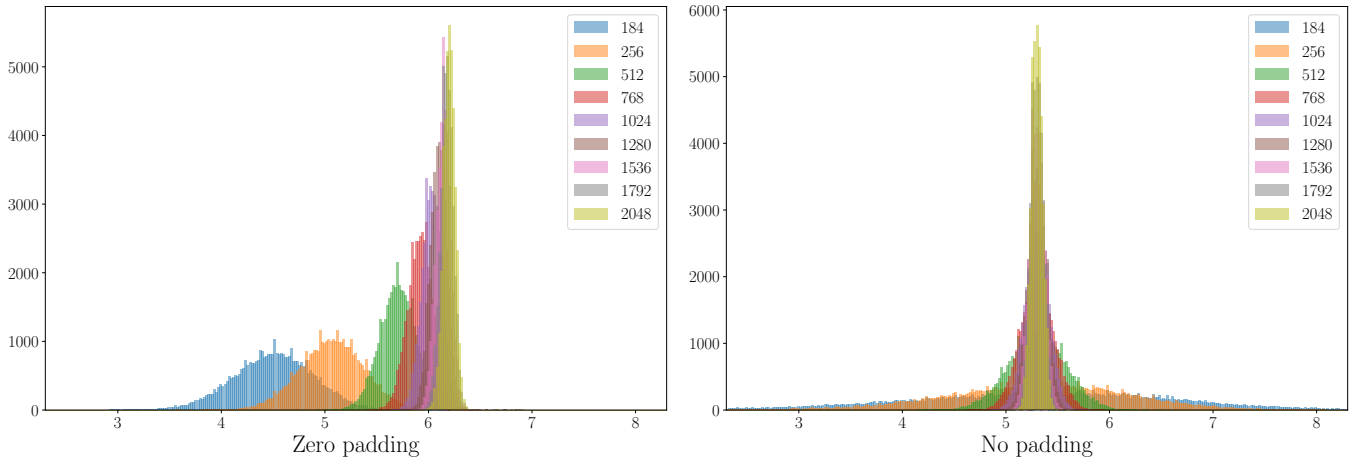

Figure 5: Effect of zero-padding on histogram of eSRNet's logits for cover images of sizes $N \times N$.

and trained to detect LSBM with 0.3 bpc as described in Section III-B.

To better understand the logits of our detector, we start visually inspecting its response on cover images of different sizes. For such inspection, we only use the testing images from our database. We can observe two phenomena from Figure 5 (left): with increasing image size, 1) the mean of the distribution increases, and 2) the variance decreases. While the decrease in variance is rather expected with more data, the shift in the mean is quite surprising. During our investigation, we found out that the only thing that could cause the mean shift was the feature poisoning due to padding.

A. Feature Poisoning Effect

We experimented with several different ways of padding in the CNN, such as mirror padding, zero padding, circular padding, and for every one we trained one eSRNet. Through an inspection of its logits across image sizes, we observed that while some methods were causing the mean-shift in the logit distributions to be smaller, it was always present.

We thus decided to disable the padding completely in the network. To preserve the flow in the residual layers of the network with skip connections, we crop appropriately the features in the skip connections. One could argue that this leads to information loss, however, we believe that the necessary information would be preserved in the main branch of the layer. The effect of disabling the padding on the logit histogram can be seen in Figure 5 (right). As we can see, for the network trained this way, we cannot see any mean-shift anymore, even at a price of higher variance for the smaller sizes. We explain this fact by reducing the feature size before the GAP. Indeed for an image of size 184×184 (256×256), the pre-GAP feature size is 12×12 (16×16) with padding and 1×1 (5×5) without padding.

While the shift in the mean can be estimated from controlled images, we find this rather impractical. For a given image under investigation, one would have to generate a dataset of images of the same size, which could be computationally very costly, especially for large images with millions of pixels. For the rest of the paper, we will only use the eSRNet trained without padding to evaluate our proposed methodology.

Note that the prior work using the MLP [48] did not consider different padding strategies so we kept the TD's implicit zero padding.

B. Scaling of Variance

Now that we see that the logit distribution mean remains unchanged across different image sizes, it is time to investigate the behavior of its variance. To this end, we state the following theorem.

Theorem 1 (Variance Scaling Law)**.** *Assume a fixed JPEG compressor compressing images with QF 100, a convolutional neural network without padding, with receptive field size R and stride S, trained on rounding errors $\mathbf{e}$ of cover images and non-adaptive stego images. Let ϕ be the detector's logit from a cover image with $N = N_1 \times N_2$ pixels. Denote $N_G = n_1^{(G)} \times n_2^{(G)}$ the pre-GAP feature size, where $n_1^{(G)} = \lceil (N_1 - R + 1)/S \rceil$ and $n_2^{(G)} = \lceil (N_2 - R + 1)/S \rceil$.*

Then $\phi \sim \mathcal{N}(\mu, \sigma_N^2)$, where $\mu \in \mathbb{R}$ is a constant and σ_N^2 decreases with N. Specifically,

if $N_1, N_2 \geq 2R$, then there exist $a > 0, b \geq 0$ such that

$$\sigma_N^2 = \frac{a}{N_G} + b. \tag{10}$$

Otherwise, for $N_1 \leq N_2$, there exist $a, c > 0, b \geq 0$

$$\sigma_N^2 = \frac{a}{N_G} + b + \frac{c n_1}{N_G} \left(1 + \psi\left(\frac{n_2 + 1}{2} \right) - \psi\left(\frac{n_1 + 1}{2} \right) \right), \tag{11}$$

where $n_1 = \min\{n_1^{(G)}, \lceil \frac{R}{S} \rceil\}$, $n_2 = \min\{n_2^{(G)}, \lceil \frac{R}{S} \rceil\}$, and $\psi(z) = \frac{\Gamma'(z)}{\Gamma(z)}$ is the digamma function.

Moreover, the absolute term b depends on the over-parametrization of the given architecture.

We refer the reader for the proof of Theorem 1 to the Appendix. We want to point out that the value μ is linked

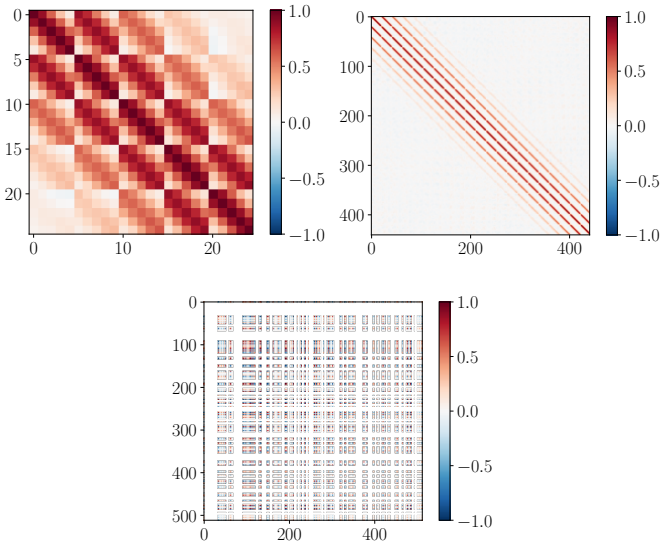

Figure 6: Top: Correlation matrix of pre-GAP features across 1000 images from a single channel. Images of size 256×256 (left) and 512×512 (right). Bottom: Correlation matrix between GAP features across 1000 images of size 512×512. The values are cropped for better visualization.

to the detectability of a given steganographic scheme, especially the embedded payload, however, a more detailed investigation is beyond the scope of this work. Moreover, the parameters a, b are specific to a given source of images (JPEG compressor) and a CNN architecture. While the assumptions in Theorem 1 are tailored to our specific steganalysis scenario, the results hold for any imaging problem where the two classes carry the same distribution everywhere, independently of the location in the image (see observations 2) and 3) in the proof). Consequently, we find the limiting behavior of the logit variance.

Corollary 2. *For $N \to \infty$, there exists $b \geq 0$ such that*

$$\sigma_N^2 \to b. \tag{12}$$

To experimentally verify Theorem 1, we computed the logits from all the training sizes $\mathcal{S}_{train}$ and computed their respective empirical variances $\bar{\sigma}_N^2$. We then found the parameters a_0, b_0 and a_1, b_1, c_1 by minimizing the mean relative absolute difference for the training image sizes

$$a_0, b_0 = \arg\min \sum_{\substack{\sqrt{N} \in \mathcal{S}_{train} \\ \sqrt{N} \geq 2R}} \frac{\left|\bar{\sigma}_N^2 - \sigma_N^2\right|}{\bar{\sigma}_N^2}, \tag{13}$$

$$a_1, b_1, c_1 = \arg\min \sum_{\substack{\sqrt{N} \in \mathcal{S}_{train} \\ \sqrt{N} \leq 2R}} \frac{\left|\bar{\sigma}_N^2 - \sigma_N^2\right|}{\bar{\sigma}_N^2}. \tag{14}$$

Since for SRNet $2R = 358$, we added images of size 360×360 to the training sizes $\mathcal{S}_{train}$ as it is the closest size of a valid JPEG image.

Solving equations (13),(14) with our dataset gives $a_0 \approx 36.5, b_0 \approx 2 \times 10^{-4}, a_1 \approx -4, b_1 \approx -0.26, c \approx 5.9$. We

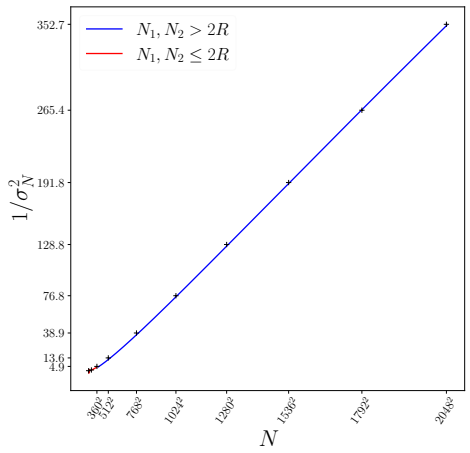

Figure 7: Logit variance as a function of the number of pixels N.

show in Figure 7 the estimated and empirical reciprocal variances w.r.t. the number of pixels N.

Although, the absolute term b in Eq. (12) can be unexpected at first, it is not that surprising. Since the eSRNet has more than 4.5M parameters, it is extremely over-parametrized for the given task. We performed the Principal Component Analysis of the GAP features[5] and found that 98% of information can be contained in only 2 channels out of 512. We conclude that many of the convolutional kernels in different channels (within the same layer) are almost identical. As a result, the features deep in the network are greatly correlated across channels giving rise to the absolute term. See Figure 6 (bottom) for the correlation matrix of the 512-dimensional GAP feature. If the network offered no redundancy, we expect the features to not be correlated across channels, which would result in $b = 0$. This observation is also supported in the proof of the theorem (see Eq. (19)). However, compressing the architecture to avoid redundancy on a given task is out of the scope of this work and we plan to address this phenomenon in the future.

As explained in the Appendix, the terms a arises from the spatial partial correlation among the pre-GAP features (and their variance), which is, in turn, caused by overlapping receptive fields. In general, the closest odd multiple of $[R/S]$ features are correlated in every direction. The partial correlations in a single channel are shown in the correlation matrices in Figure 6 (top). Finally, for images where one of the sizes is smaller than $2R$, all the pre-GAP features are horizontally correlated if the image width is smaller than $2R$ and/or vertically correlated if the image height is smaller than $2R$, which leads to the term c.

Based on the above discussion, we can conclude that to reduce the logit variance as much as possible (leading to better class separation), the CNN architecture should 1) have a small ratio R/S to reduce the spatial partial correlations among neighboring pre-GAP features, 2) have

[5]The feature vector resulting from the Global Average Pooling.

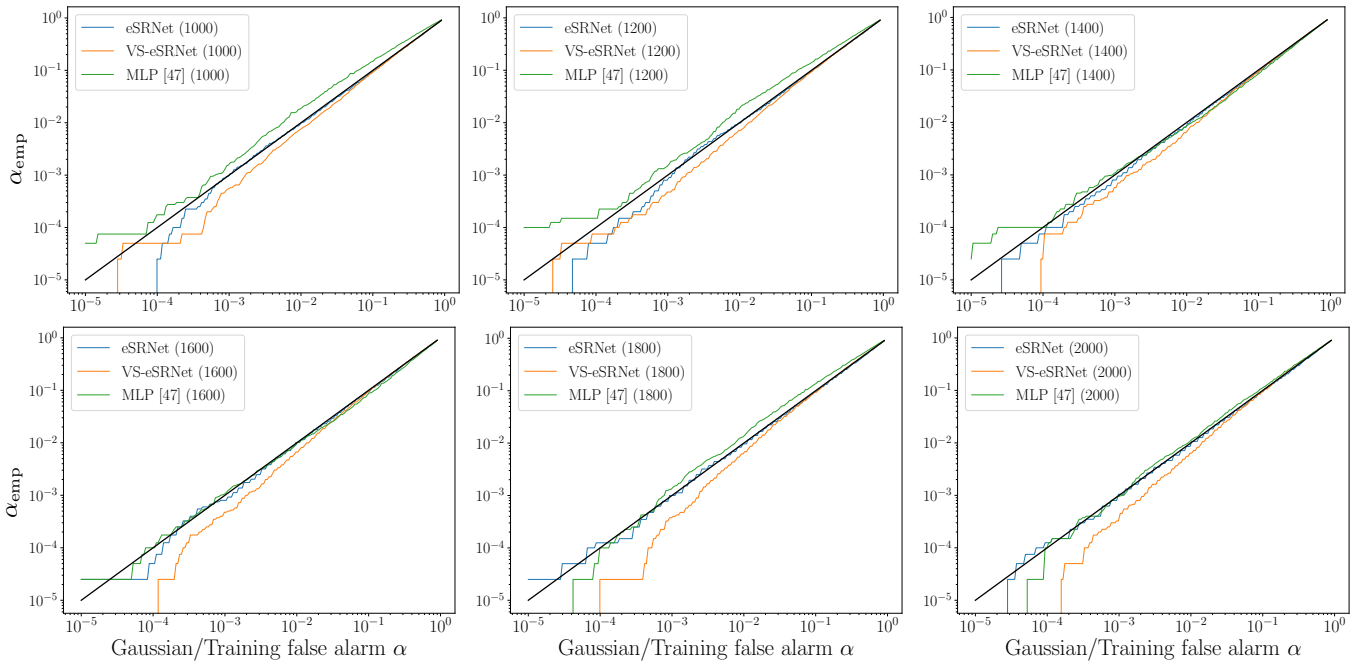

Figure 8: Comparisons between prescribed FPR α and practical FPR α_{emp} for the different schemes and images of sizes $N \times N$ (N is listed in the legend). The MLP's α_{emp} is shown as a function of the empirical FP rate on its training images of the closest size in $\mathcal{S}_{train}$.

a small receptive field size R to prevent spatial correlations among all pre-GAP features for images of size smaller than $2R$, and 3) be pruned to avoid any redundancy in order to eliminate the absolute term b. While the pruning can always be performed as a last step of the detector development, the other two points depend on the specific requirements of a steganalyst. For instance, if we only aim to steganalyze images of size at least 512×512, it is sufficient to consider architectures with $R < 256$.

C. Gaussianity of Logits and FPR control

The last step of our logit investigation deals with the gaussianity of the distribution. Indeed, having the mean and variance of the distribution would be rather useless, unless we know which type of distribution we are dealing with since ultimately we aim to use this distribution to prescribe a threshold for a given FP rate. Although there are various ways of verifying gaussianity, we use the following approach. For a given testing image size in $\mathcal{S}_{test}$, we compute the mean μ as the average mean of the logit distributions for image sizes in $\mathcal{S}_{train}$ and the logit variance σ_N^2 using Eq. (10) with a, b from the previous section. For a prescribed (Gaussian) FP rate α, we compute the Gaussian threshold as

$$T_\alpha = \mu + \sigma_N Q^{-1}(\alpha), \tag{15}$$

where Q^{-1} is the inverse Q-function.

We then use the same threshold to compute the empirical FP rate α_{emp} from the logits. The eSRNet's comparison of α and α_{emp} is shown in Figure 8. For all the testing sizes in $\mathcal{S}_{test}$, we can see a very clear match between the prescribed Gaussian FP rates and the empirical ones even for the values close to 10^{-4}. Not only does this strategy verify the gaussianity of the distribution, but it also immediately demonstrates the adequation between the prescribed FP rates and the practical ones for different image sizes that do not belong to the training set. We cannot experimentally verify the relation for smaller FP rates due to the limited testing dataset.

With an accurate model of the cover class, we can use this detector as a one-class classifier, potentially detecting previously unseen steganography. We verify this statement in the next section by steganalyzing two content-adaptive steganographic algorithms.

V. Results and comparisons

We compare here three methods that perform steganalysis for different image sizes:

- eSRNet, where the training is performed only on 512×512 images but the distributions of the logits (see section IV-B) are derived from the different sizes provided in the training set.
- VS-eSRNet, where both the training and the estimation of the logit distributions are performed with the variable image sizes provided in the training set.
- the MLP (see I-A, [24]), which links the decision threshold to the FPR by resorting to Monte-Carlo methods. Note that one drawback of this methodology is that the prescribed FPR relies on the size of the training set, e.g. for a FPR of 10^{-n} as a rule of thumb 10^{n+1} images are necessary to obtain an accurate estimate of the FPR.

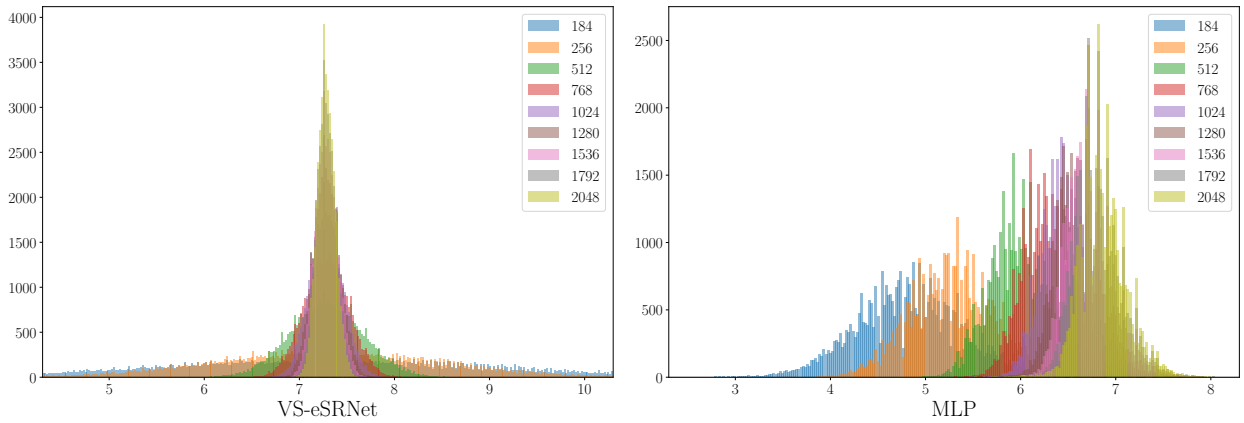

Figure 9: Histograms of the VS-eSRNet's and MLP's logits for cover images of size $N \times N$.

To compare these methods, we consider two criteria: the quality of the prescribed FP rate and the probability of error (1) on unseen steganography. First, we focus on the the VS-eSRNet. Figure 9 (left) shows the logits coming from this detector and we can see that even in this case, the mean of the distribution is independent of the image size. However, we see that even in this case, the variance of the logits changes w.r.t. the image size. We thus used the same methodology as for eSRNet, proposed in the previous section. In particular, we estimated the scaling parameters a, b according to equation (13). Furthermore, by comparing with Figure 5, we can notice that the mean is larger than that of eSRNet, but the variance also seems to be bigger. We attribute this to the more diverse training cover set but also the stego set, because we did not adjust the embedding payload to maintain the same detectability according to the Square Root Law [32]. Consequently, we can see in Table I that the probability of error on LSBM images is much smaller on the smaller payload for this detector that has seen a wider variety of LSBM-embedded images during training. However, results in Tables II and III show that the performance of these two detectors is very comparable on previously unseen, content-adaptive UERD and J-UNIWARD steganographic algorithms. Yet another difference is shown in Figure 8 in which we see that the prescribed FP rates start to deviate for values below 10^{-2}, suggesting that the left tail of the distribution is thicker than that of a Gaussian distribution. Overall, the VS-eSRNet only provides better performance on seen steganography, but the prescribed FP rates are not as reliable as those of e-SRNet. Moreover, the computational cost increases rapidly, requiring the use of 4 GPUs during its training.

By inspecting the logits of the MLP detector (see Figure 9 (right)), we see that the distribution changes its mean across image sizes because of the implicit padding in the tile detector, therefore we cannot prescribe a theoretical FP rate since that would require creating a dataset for every image size under investigation in order to estimate the distribution mean. Instead, for a given testing image, we take the logits of training images of the closest size in $\mathcal{S}_{train}$. With these training logits, we then empirically find a threshold that corresponds to a prescribed FP rate and use it to compute the empirical FP rate α_{emp} on the testing images. The comparison in Figure 8 indicates that this works quite well, however, there is one major drawback. For images of size bigger than 2048×2048, we would have to use the largest images in the training database to prescribe a decision threshold, which could lead to severe underperformance. A similar problem would arise if we were to test an image of a size smaller than in the training data. To demonstrate this issue, we show in Figure 10 α_{emp} for images of size 1000 and 2000 if the training data was not optimal, e.g. $\mathcal{S}^{(1)}_{train} = \{1280, 1536, 1792, 2048\}$, and $\mathcal{S}^{(2)}_{train} = \{184, 256, 512, 768, 1024\}$ respectively.

Finally, Tables I, II, and III show that the performance of the MLP detector is inferior to the other two proposed methods, by up to 40% in terms of P_{E} for the largest LSBM images at the smallest payload 0.01 bpc. We believe this is due to the MLP overfitting on the larger training payload 0.3 bpc. Consequently, our approach outperforms this state-of-the-art method regarding both, the detection

Table I: P_{E} for LSBM images of different size $N \times N$.

bpc	Method	N					
		1000	1200	1400	1600	1800	2000
0.01	eSRNet	0.17	0.12	0.08	0.05	0.03	0.02
	VS-eSRNet	**0.10**	**0.06**	**0.04**	**0.02**	**0.02**	**0.01**
	MLP [48]	0.43	0.43	0.42	0.41	0.41	0.41
0.05	eSRNet	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	VS-eSRNet	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	MLP [48]	0.08	0.06	0.04	0.03	0.03	0.02

Table II: P_{E} for unseen UERD images of different sizes $N \times N$ (bold/minimum scores take into account the value before rounding).

bpc	Method	N					
		1000	1200	1400	1600	1800	2000
0.01	eSRNet	0.15	0.11	0.07	0.04	**0.02**	**0.01**
	VS-eSRNet	**0.14**	**0.10**	**0.06**	**0.04**	0.02	0.01
	MLP [48]	0.29	0.25	0.22	0.19	0.17	0.15
0.05	eSRNet	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	VS-eSRNet	**0.01**	0.01	0.00	0.00	**0.00**	**0.00**
	MLP [48]	0.02	0.01	0.00	0.00	0.00	0.00

Table III: P_{E} for unseen J-UNIWARD images of different sizes $N \times N$ (bold/minimum scores take into account the value before rounding).

bpc	Method	N					
		1000	1200	1400	1600	1800	2000
0.01	eSRNet	0.08	0.04	0.02	0.01	0.00	0.00
	VS-eSRNet	**0.05**	**0.02**	**0.01**	**0.00**	**0.00**	**0.00**
	MLP [48]	0.40	0.39	0.39	0.38	0.38	0.37
0.05	eSRNet	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	VS-eSRNet	0.00	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	MLP [48]	0.03	0.02	0.01	0.01	0.01	0.01

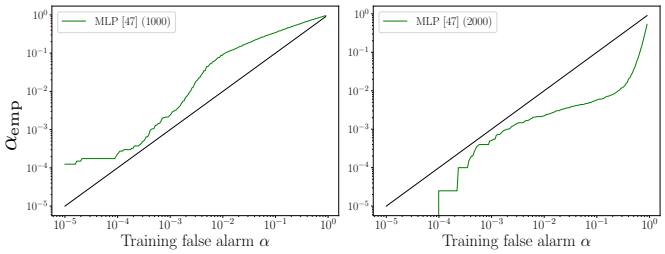

Figure 10: The MLP's α_{emp} as a function of the empirical FP rate α on insufficient training sizes. Left: testing images of size 1000×1000 while training image sizes are $\mathcal{S}^{(1)}_{train}$, right: testing images of size 2000×2000 while training image sizes are $\mathcal{S}^{(2)}_{train}$. In both cases, training images of the closest size were used to prescribe a decision threshold for a desired α, i.e. 1280×1280 and 1024×1024.

performance and the FPR control.

Note that while the probability of error is generally smaller on UERD images for eSRNet and MLP detectors than on LSBM images, it is the opposite for the VS-eSRNet. We attribute this to the more diverse stego training set of the VS-eSRNet, which causes it to overspecialize on the LSBM images. Furthermore, since the UERD and J-UNIWARD algorithms are content-adaptive, they create more embedding changes than the non-adaptive LSBM, giving more evidence to the detector that the images do not belong to the cover class.

VI. Conclusions

In this work, we introduced, for the first time, the concept of size-independent steganalysis with control over the false positive rate of a deep learning detector based on the RJCA. We achieve this by modeling a deep learning detector's cover class soft outputs (logits) with a Gaussian distribution. In consequence, we can use the detector as a one-class classifier separating cover images from stego images. This detector achieves state-of-the-art steganalysis performance (even on unseen steganography) and FP rate control for images of arbitrary size.

First, we showed that padding inside the convolutional layers of a CNN detector negatively affects the distribution of the input image which causes a mean-shift of the logit distribution. With the padding disabled, we proved a theorem stating that the variance of the logit distribution can be expressed as an affine function of the reciprocal number of pixels. With a detector trained on images of a fixed size, the e-SRNet, we estimated the parameters of this affine relation from several predefined datasets of images of different sizes and were able to theoretically prescribe a decision threshold for a desired false alarm rate for images of any size. We then experimentally verified that the empirical false alarm rate closely matches the prescribed one. Finally, we verified that the classifier generalizes well to detect other unseen steganography with much smaller embedding payloads than those used during the detector training.

By relying solely on a deep learning approach - providing images of different sizes during the training - we have seen that the logits even from this detector, the VS-eSRNet, follow the same trend as if trained on fixed-size images only. Using the proposed methodology for this detector revealed that the performance of the two detectors is rather similar. However training the VS-eSRNet requires more computational resources and the resulting cover logit distributions have a thicker left tail, causing discrepancies for false positive rates below 10^{-2}.

While we limited ourselves only to JPEG images compressed with the highest quality, in order to exploit the decompression rounding errors, we believe that the same methodology can be applied to more classical spatial deep learning steganalysis that works on image pixels. This would allow us to steganalyze JPEG images of lower quality and perhaps even spatial domain steganography. While the padding will have a similar, perhaps even smaller, effect on the logit distribution in such a scenario, the image content and content-adaptive steganography will have a much stronger impact on the distribution.

Acknowledgement

This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011012855 made by GENCI. This work received funding from the European Union'Horizon 2020 research and innovation program under grant agreement No 101021687 (project "UNCOVER").

Appendix
Proof of Theorem 1

Let ϕ be the logits of the cover class coming from a CNN with receptive field R and stride S and without any padding. Let $X_{c,i}$ be the vectorized pre-GAP feature entering the Global Average Pooling (GAP) at spatial position $i \in \{1, \dots, N_G\}$ in a channel $c \in \{1, \dots, C\}$, where N_G is the number of features before GAP and C is the number of filters (channels) in the last convolutional layer ($C = 512$ is for SRNet, $C = 1280$ for EfficientNet-B0). Let us recall from Section II-B1 that

$$N_G = n_1^{(G)} \times n_2^{(G)}.$$

Let further $\mathbf{G} \in \mathbb{R}$ be the vector of GAP features and $\mathbf{w} \in \mathbb{R}^{C\times 1}$ be the (cover) weight vector from the fully-connected layer. We can express the cover logit as

$$\begin{aligned}\phi &= \mathbf{G}\cdot\mathbf{w}\\ &= \sum_{c=1}^{C} w_c G_c\\ &= \frac{1}{N_G}\sum_{c=1}^{C} w_c \sum_{i=1}^{N_G} X_{c,i}\end{aligned}$$

Let us now make three key observations:

1) Convolutional neural networks are shift equivariant [16], [38],

2) For a given JPEG compressor, the image content under scrutiny (the rounding errors e_{ij}) follows the same distribution independently of the position in the image (with some exceptions in constant blocks [8]),

3) Non-adaptive steganography changes the stego distribution in the same way, independently of the position in the image. We want to point out that the non-adaptivness of steganography is only required during the training of the detector.

These three observations allow us to make the following conclusions. For any channel $c \in \{1,\dots,C\}$, all the cover pre-GAP features have the same mean and variance

$$\mathrm{Var}\left(X_{c,i}\right) = \sigma_c^2, \tag{16}$$

$$\mathbb{E}\left[X_{c,i}\right] = \mathbb{E}\left[X_c\right], \quad \forall i = 1,\dots,N_G. \tag{17}$$

In the following, we will be dropping the channel index for better readability. We note that due to the overlap of the receptive fields in the network, each row (column) of the covariance matrix of $[X_1,\dots,X_{N_G}]$ contains $n_{R,S}$ positive correlations and $N_G - n_{R,S}$ features that are uncorrelated, where $n_{R,S}$ is the number of overlapping receptive fields on a single pixel: $n_{R,S} = n_1 \times n_2$.[6] The above formula dictates that if the image is too small, all the features will be correlated. In other words, if $n_1, n_2 \le \frac{R}{S}$, which is equivalent to $N_1, N_2 \le 2R-1$, we obtain $n_{R,S} = N_G$. We now consider different situations, depending on the values n_1, n_2.

First, let $n_1 \le n_2 \le \frac{R}{S}$. We will model the covariance between neighboring features as

$$\mathrm{Cov}\left(X_i, X_j\right) = \frac{b}{d_1(X_i, X_j)},$$

where $d_1(X_i, X_j)$ is the ℓ_1 distance between the spatial position of X_i and X_j because the overlap between neighboring receptive fields decreases linearly in both directions. As such, summing the covariances of correlated features can be expressed as summing the radii of $4r$ points on concentric circles with radii $r \in \{1,\dots,\frac{n_1}{2}\}$. Additionally, we have to consider the points on 'circles' having one of the axes longer than the other, if $n_1 < n_2$. To sum a single (any) row of the covariance matrix, we compute

$$\begin{aligned}\sum_j \mathrm{Cov}(X_i, X_j) &= \sigma^2 + b\sum_{r=1}^{(n_1-1)/2}\frac{4r}{r}\\ &\quad + b\sum_{r=(n_1+1)/2}^{(n_2-1)/2}\frac{4(\frac{n_1-1}{2}+1)-2}{r}\\ &= \sigma^2 + 2b(n_1-1)\\ &\quad + 2n_1 b\left(\psi\left(\frac{n_2+1}{2}\right) - \psi\left(\frac{n_1+1}{2}\right)\right),\end{aligned}$$

where $\psi(z) \sim \ln(z) - \frac{1}{2z}$ is the digamma function. After some simplifications, we can write

$$\begin{aligned}&\sum_j \mathrm{Cov}(X_i, X_j) =\\ &\qquad a + bn_1\left(1 + \psi\left(\frac{n_2+1}{2}\right) - \psi\left(\frac{n_1+1}{2}\right)\right).\end{aligned} \tag{18}$$

In the case of $n_1 \le \frac{R}{S} \le n_2$, we simply substitute n_2 with $\frac{R}{S}$ in Eq. (18).

Let us assume for the rest of the proof that $n_1, n_2 \ge \frac{R}{S}$ ($N_1, N_2 \ge 2R$). Note that this is equivalent to substituting both n_1 and n_2 with $\frac{R}{S}$ in Eq. (18). The rest of the proof for the other cases considered above would proceed in the same manner.

Since $n_{R,S} = \left[\frac{R}{S}\right]^2$, the partial correlations are constant w.r.t. the image size, and thus we can write

$$\begin{aligned}\mathrm{Var}\left(\sum_i X_i\right) &= \sum_{i,j}\mathrm{Cov}\left(X_i, X_j\right)\\ &= N_G\left(\sigma^2 + b\left(n_{R,S}-1\right)\right)\\ &= a_0 N_G.\end{aligned}$$

It follows that

$$\begin{aligned}\mathrm{Var}\left(G\right) &= \frac{1}{N_G^2}\mathrm{Var}\left(\sum_i X_i\right)\\ &= \frac{a_0}{N_G}.\end{aligned}$$

The variance of the logit is

$$\begin{aligned}\sigma_N^2 &= \mathrm{Var}\left(\sum_{c=1}^{C} w_c G_c\right)\\ &= \sum_{c,d=1}^{C} w_c w_d \mathrm{Cov}\left(G_c, G_d\right)\\ &= \frac{\|\mathbf{w}\|^2 a_0}{N_G} + \sum_{c=1}^{C}\sum_{d\neq c} w_c w_d \mathrm{Cov}\left(G_c, G_d\right)\\ &= \frac{a}{N_G} + b,\end{aligned} \tag{19}$$

where the constant $b \in \mathbb{R}$ depends on the correlations across channels caused by over-parametrization of the architecture on a given task. Note that we were able to

[6] In our experiments with SRNet and images of size $N \times N$, $N > 2R = 358$, we observed that $n_{R,S} = 11 \times 11$, which corresponds to the value $[R/S]^2$.

simplify the expression since $\mathrm{Cov}(G_c, G_d)$ does not depend on N_G for $c \neq d$.

It remains to prove that the mean of the logit does not change across different image sizes. We now express the mean as

$$\mu = \frac{1}{N_G} \sum_{c=1}^{C} w_c \sum_{i=1}^{N_G} \mathbb{E}\left[X_{c,i}\right],$$

and remind the reader that the mean in a channel is the same for every spatial position (see Eq. (17)). It follows that

$$\mu = \sum_{c=1}^{C} w_c \mathbb{E}\left[X_c\right],$$

which is independent of the image size. □

References

[1] A. Araujo, W. Norris, and J. Sim. Computing receptive fields of convolutional neural networks. *Distill*, 2019. https://distill.pub/2019/computing-receptive-fields.

[2] S. Bernard, T. Pevný, P. Bas, and J. Klein. Exploiting adversarial embeddings for better 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.

[3] R. Böhme and A. Westfeld. Breaking Cauchy model-based JPEG steganography with first order statistics. In P. Samarati, P. Y. A. Ryan, D. Gollmann, and R. Molva, editors, *Computer Security - ESORICS 2004. Proceedings 9th European Symposium on Research in Computer Security*, volume 3193 of Lecture Notes in Computer Science, pages 125–140, Sophia Antipolis, France, September 13–15, 2004. Springer, Berlin.

[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 P. Bas. Fighting the reverse jpeg compatibility attack: Pick your side. In *The 10th ACM Workshop on Information Hiding and Multimedia Security*, Santa Barbara, CA, June 27–28, 2022.

[6] J. Butora and P. Bas. Side-informed steganography for jpeg images by modeling decompressed images. *IEEE Transactions on Information Forensics and Security*, 18:2683–2695, 2023.

[7] J. Butora, P. Bas, and R. Cogranne. Analysis and mitigation of the false alarms of the reverse jpeg compatibility attack. In D. Moreira, editor, *The 11th ACM Workshop on Information Hiding and Multimedia Security*, Chicago, IL, June 28–30, 2023. ACM Press.

[8] J. Butora and J. Fridrich. Reverse JPEG compatibility attack. *IEEE Transactions on Information Forensics and Security*, 15:1444–1454, 2020.

[9] J. Butora and J. Fridrich. Extending the reverse JPEG compatibility attack to double compressed images. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Canada, June 6–11, 2021.

[10] J. Butora, Y. Yousfi, and J. Fridrich. How to pretrain for steganalysis. In *The 9th ACM Workshop on Information Hiding and Multimedia Security*, Brussels, Belgium, June 21–25, 2021.

[11] R. Cogranne. Selection-channel-aware reverse JPEG compatibility for highly reliable steganalysis of JPEG images. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, pages 2772–2776, Barcelona, Spain, May 4–8, 2020.

[12] 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.

[13] 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.

[14] R. Cogranne, Q. Giboulot, and P. Bas. Efficient steganography in jpeg images by minimizing performance of optimal detector. *IEEE Transactions on Information Forensics and Security*, 17:1328–1343, 2022.

[15] R. Cogranne and F. Retraint. An asymptotically uniformly most powerful test for LSB Matching detection. *IEEE Transactions on Information Forensics and Security*, 8(3):464–476, 2013.

[16] T. Cohen and M. Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR.

[17] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pages 248–255, June 20–25, 2009.

[18] S. Dumitrescu, X. Wu, and N. D. Memon. On steganalysis of random LSB embedding in continuous-tone images. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2002*, pages 324–339, Rochester, NY, September 22–25, 2002.

[19] S. Dumitrescu, X. Wu, and Z. Wang. Detection of LSB steganography via Sample Pairs Analysis. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of Lecture Notes in Computer Science, pages 355–372, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[20] E.Levecque, J. Butora, J. Klein, and P. Bas. Vers une steganalyse certifiée pour des images JPEG. In *XXIXème Colloque Francophone de Traitement du Signal et des Images*, Nancy, France, September 6–9 2022.

[21] E.Levecque, J. Klein, P. Bas, and J. Butora. Toward reliable jpeg steganalysis (at qf100). In *IEEE International Workshop on Information Forensics and Security*, Shanghai, China, December 12–16 2022.

[22] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications.* Cambridge University Press, 2009.

[23] 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.

[24] 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.

[25] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, pages 1–77, 2023.

[26] Q. Giboulot, R. Cogranne, D. Borghys, and P. Bas. Effects and Solutions of Cover-Source Mismatch in Image Steganalysis. *Signal Processing: Image Communication*, August 2020.

[27] L. Guo, J. Ni, W. Su, C. Tang, and Y. Shi. Using statistical image model for JPEG steganography: Uniform embedding revisited. *IEEE Transactions on Information Forensics and Security*, 10(12):2669–2680, 2015.

[28] 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.

[29] 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.

[30] A. D. Ker. Quantitative evaluation of pairs and RS steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 83–97, San Jose, CA, January 19–22, 2004.

[31] A. D. Ker. Batch steganography and the threshold game. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Elec-*

tronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, volume 6505, pages 04 1–13, San Jose, CA, January 29–February 1, 2007.

[32] A. D. Ker. The square root law of steganography. 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. ACM Press.

[33] A. D. Ker and T. Pevný. A mishmash of methods for mitigating the model mismatch mess. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2014*, volume 9028, pages 1601–1615, San Francisco, CA, February 3–5, 2014.

[34] J. Kodovský and J. Fridrich. Steganalysis of JPEG images using rich models. In A. Alattar, N. D. Memon, and E. J. Delp, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2012*, volume 8303, pages 0A 1–13, San Francisco, CA, January 23–26, 2012.

[35] J. Kodovský, V. Sedighi, and J. Fridrich. Study of cover source mismatch in steganalysis and ways to mitigate its impact. In A. M. Alattar, N. D. Memon, and C. D. Heitzenrater, editors, *Media Watermarking, Security, and Forensics 2014*, volume 9028, pages 204 – 215. International Society for Optics and Photonics, SPIE, 2014.

[36] E.L. Lehmann and J.P. Romano. *Testing Statistical Hypotheses, Second Edition*. Springer, 3rd edition, 2005.

[37] 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.

[38] N. McGreivy and A. Hakim. Convolutional layers are not translation equivariant. *arXiv preprint arXiv:2206.04979*, 2022.

[39] T. Mingxing and V. L. Quoc. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, volume 97, pages 6105–6114, June 9–15, 2019.

[40] 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, 2015.

[41] T. Thai, R. Cogranne, and F. Retraint. Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm. *Image Processing, IEEE Transactions on Image Processing*, 23(5):1–14, May 2014.

[42] T. H. Thai, R. Cogranne, and F. Retraint. Optimal detection of OutGuess using an accurate model of DCT coefficients. In *Sixth IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, December 3–5, 2014.

[43] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In A. Pfitzmann, editor, *Information Hiding, 3rd International Workshop*, volume 1768 of Lecture Notes in Computer Science, pages 61–75, Dresden, Germany, September 29–October 1, 1999. Springer-Verlag, New York.

[44] 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.

[45] 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.

[46] 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.

[47] Y. Yousfi, J. Butora, J. Fridrich, and C. F. Tsang. Improving EfficientNet for JPEG steganalysis. In *The 9th ACM Workshop on Information Hiding and Multimedia Security*, Brussels, Belgium, June 21–25, 2021.

[48] 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.

[49] 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.

[50] X. Zhang, X. Kong, P. Wang, and B. Wang. Cover-source mismatch in deep spatial steganalysis. In H. Wang X. Zhao, Y. Shi, H. J. Kim, and A. Piva, editors, *Digital Forensics and Watermarking*, pages 71–83. Springer International Publishing, 2020.