CNN Steganalyzers Leverage Local Embedding Artifacts

Yassine Yousfi
Department of ECE
Binghamton University
yyousfi1@binghamton.edu

Jan Butora
Department of ECE
Binghamton University
jbutora1@binghamton.edu

Jessica Fridrich
Department of ECE
Binghamton University
fridrich@binghamton.edu

***Abstract*—While convolutional neural networks have firmly established themselves as the superior steganography detectors, little human-interpretable feedback to the steganographer as to how the network reaches its decision has so far been obtained from trained models. The folklore has it that, unlike rich models, which rely on global statistics, CNNs can leverage spatially localized signals. In this paper, we adapt existing attribution tools, such as Integrated Gradients and Last Activation Maps, to show that CNNs can indeed find overwhelming evidence for steganography from a few highly localized embedding artifacts. We look at the nature of these artifacts via case studies of both modern content-adaptive and older steganographic algorithms. The main culprit is linked to "content creating changes" when the magnitude of a DCT coefficient is increased (Jsteg, –F5), which can be especially detectable for high frequency DCT modes that were originally zeros (J-MiPOD). In contrast, J-UNIWARD introduces the smallest number of locally detectable embedding artifacts among all tested algorithms. Moreover, we find examples of inhibition that facilitate distinguishing between the selection channels of stego algorithms in a multi-class detector. The authors believe that identifying and characterizing local embedding artifacts provides useful feedback for future design of steganographic schemes.**

***Index Terms*—Steganalysis, CNN, LDEA, explainable machine learning**

I. Introduction

Recently, steganalysis has undergone an explosive development due to employment of deep convolutional neural networks (CNNs) [1]. Major improvements in detection accuracy have been achieved for all embedding algorithms and both domains. Immediately, speculations appeared about why these detectors perform so much better than classifiers trained on high-dimensional rich media models. The usual explanation is the network's ability to jointly optimize the image representation ("feature formation") as well as the classifier. Indeed, to keep the dimensionality of co-occurrences from which rich models are built reasonably low, noise residuals need to be harshly truncated and quantized. Furthermore, training on large datasets becomes computationally infeasible even with low-complexity classifiers [2], [3].

There is one more fundamental difference between CNN detectors and rich models. The latter are by their construction macroscopic quantities of local statistics collected in a global fashion from the entire image. Rich models are essentially collections of histograms. This limits them to being predominantly "integrators" of local embedding traces across the image that achieve non-trivial detection power by leveraging some form of the Central Limit Theorem (CLT). In contrast, CNNs do not natively form histograms, and instead process the outputs of convolutions or "noise residuals" in a different fashion that is believed to allow both integration as well as detection of localized embedding traces. To the best knowledge of the authors, however, no study has been put forward that would present evidence for this claim.

Aided with visualization tools, we argue that CNN detectors leverage Locally Detectable Embedding Artifacts (LDEAs) in their decision making. Leaving a few stego blocks with LDEAs in the stego image is enough for a reliable detection even with other CNN architectures. In contrast, rich models are not necessarily able to correctly classify all stego images with LDEAs. By taking a closer look at modern content-adaptive algorithms J-MiPOD [4] and J-UNIWARD [5], and older embedding schemes (F5 [6], –F5 [7], and Jsteg), we discover that LDEAs are mostly associated with content-creating changes when the magnitude of a DCT coefficient is increased and, especially when a high-frequency cover DCT equal to 0 is changed to a non-zero value. Additionally, we argue that LDEAs and inhibition play a role when training a multi-class detector to distinguish between selection channels of different embedding algorithms. Our findings provide valuable qualitative and human interpretable feedback to the steganographer that could be taken into consideration for design of future stego algorithms.

In the next section, we describe the datasets and detectors employed in our experiments. Section III describes visualization tools used in this work. LDEAs are defined in Section IV, which contains case studies involving JPEG steganographic algorithms. In Section V, we study a multi-class CNN distinguishing between bUERD, J-UNIWARD, and covers to demonstrate that it uses inhibitory response with LDEAs on the image boundary. Section VI concludes the paper.

II. Experimental Setting

A. Datasets and detectors

We use the ALASKA II 256 × 256 dataset [8], which contains 3×25,000 cover images compressed with quality

WIFS'2021, December 7-10, 2021, Montpellier, France. 978-1-7281-9930-6/20/$31.00 ©2021 IEEE.

factors 75, 90, and 95. The covers were randomly divided into three sets with 3×22,000, 3×1,000, and 3×2,000 images for training, validation, and testing, respectively. The images were embedded only in the luminance channel Y. The findings of this paper are consistent when using the 256×256 grayscale BOSSbase+BOWS2 cover dataset but we do not report on them due to space constraint.

EfficientNet B4 [9] was pre-trained on ImageNet [10] and refined for steganalysis in the JPEG domain [11], [12] with the same training schedule as in Section 4.2 in [12]. No modifications were done to the EfficientNet B4 architecture besides changing the original Fully Connected (FC) layer to a binary classification FC or a three-class FC in Section V. We also use the SRNet [13] trained without pair constraint as in [11] and DCTR [14] with FLD ensemble [3].

III. Toolbox

A. Integrated Gradients

Integrated Gradient (IG) [15] is a technique for computing a map describing the importance of each pixel when facing a stego image. The soft output of a CNN is a function $f : \mathbb{R}^N \to \mathbb{R}$, $N = 256 \times 256$, whose domain are 256×256 images. The cover, stego, and the baseline image are denoted, respectively, c, s, and b. The IG algorithm is a pixel attribution function:

$$\phi(f, s, b) = (s - b) \odot \int_0^1 \frac{\mathrm{d}f\,(b + \alpha(s - b))}{\mathrm{d}s}\,\mathrm{d}\alpha, \qquad (1)$$

where $\mathrm{d}f/\mathrm{d}s \in \mathbb{R}^N$ is the gradient of f w.r.t. to the input s, $\odot$ denotes element-wise multiplication, and $\phi \in \mathbb{R}^N$. This algorithm belongs to path methods [16] and satisfies some desirable properties, such as, but not limited to, linearity, symmetry preserving, and completeness $\sum_{p=1}^{N} \phi_p(f, s, b) = f(s) - f(b)$. It accumulates the gradients on convex combinations of the baseline b and the input s. This accumulation encapsulates how the network's output evolves from $f(b)$ to $f(s)$. The multiplication by $s-b$ comes from the fact that the derivative is taken with respect to the path $\gamma(\alpha) = b + \alpha(s–b)$. In practice, this multiplication can be omitted as we do in Section V. The choice of b will be discussed in Section III-A1. The integral is approximated using a Riemman sum with 100 steps and the gradient is evaluated using pytorch's automatic differentiation. We use the implementation available in the Captum library.[1]

The map $\phi(f, s, b)$, which has the same shape as the input of the CNN, 256×256, is then averaged over 8×8 non-overlapping blocks along the spatial dimensions to obtain a $256/8 \times 256/8$ block importance map $\psi_r(f, s, b)$, $r = 1, \ldots, 32 \times 32$.

1) Choice of the baseline: Top k insertion test: While the IG algorithm can use an arbitrary baseline, the cover version of the stego image is the most appropriate baseline because it relates to the concept of missingness [17]. The cover represents exactly the missing signal of interest, the stego noise. Note that when using the cover image c as a baseline, the difference from the baseline in the IG algorithm are the stego changes $s - c$ in the spatial domain. Since the cumulative gradients (1) are modulated by the stego changes, $\phi(f, s, c)$ and $\psi(f, s, c)$ are zero for pixels and blocks without any changes.

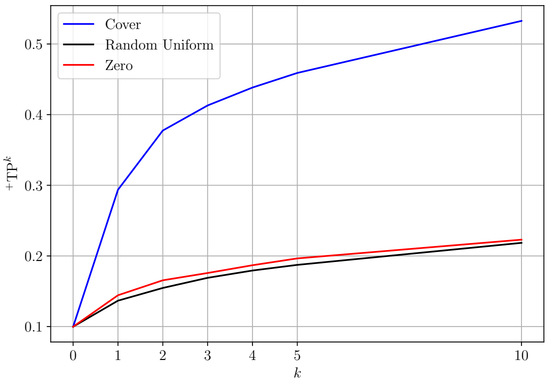


Figure 1. $^{+}\mathrm{TP}^k$ rate for covers $^{+}c^k$ with top k inserted stego blocks determined using IG with different baselines: cover, random uniform, and zero. EfficientNet B4 trained for 0.5 bpnzac J-MiPOD.

We now compare three choices for the baseline to show that the cover baseline is a suitable choice: cover, zero, and a random image with each pixel sampled independently from a uniform distribution on $[0, 1)$. For each cover-stego (c, s) pair from the test set, we compute $\psi(f, s, b)$ and identify "top" k blocks with the largest ψ that contain at least one stego change. Then, we generate from the cover c a new "stego" image, $^{+}c^k$, by only keeping the embedding changes in the top k blocks (blocks with maximal attribution ψ). The $^{+}\mathrm{TP}^k$ rate is the percentage of $^{+}c^k$ images in the test set predicted as stego using a decision threshold set for 10% False Alarm (FA) rate. Figure 1 shows the $^{+}\mathrm{TP}^k$ for EfficientNet B4 trained on 0.5 bpnzac J-MiPOD and three types of baseline images as a function of k. The cover image is clearly the best baseline for identifying the blocks that most increase the confidence of the network.

Note that even though $^{+}c^k$ are not necessarily samples of stego images (even with a lower payload), they are natural looking images, unlike insertion/ablation evaluations done in the explainable ML literature (c.f. [17]), where insertion/ablation tests are done by blurring/dropping areas of the image. This makes the inputs used in the top k insertion test fairly close to the original training distribution.

B. Last activation

In addition to IG, we use a gradient-free localization technique which we call "last activation." We essentially disable the last global pooling of a CNN and use the FC layer weights and bias as a 1×1 convolution to obtain a 16×16 matrix for the SRNet and 8×8 matrix for the EfficientNet B4 (for an input of shape 256×256). Then, we only keep the positive values in this matrix (positive logits of the stego class) and nullify the rest (rectification) to obtain a visualizable activation map. For

[1] https://github.com/pytorch/captum

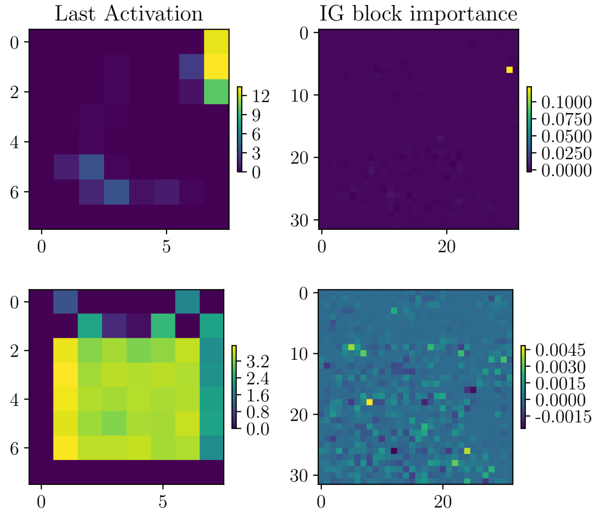


Figure 2. Last activation (left) and IG block importance map (right) of EfficientNet B4 for image '27938.jpg' embedded with J-MiPOD (top) and J-UNIWARD (bottom). Note that both images are detected as stego with $p_{\text{stego}} = 0.99$ by EfficientNet B4.

example, Figure 2 shows the last activation of EfficientNet B4 for a J-MiPOD and J-UNIWARD image and the corresponding IG block importance maps. The figure will be commented upon in more detail in Section IV-D.

IV. Locally Detectable Embedding Artifacts LDEAs

In this section, we define the concept of a Locally Detectable Embedding Artifact (LDEA) and use the tools explained above to analyze how CNNs detect selected modern content-adaptive and old steganographic methods. Three modern stego methods are included in the study: J-UNIWARD [5], J-MiPOD, and bUERD [18]. The last is a version of the UERD algorithm as implemented during ALASKA II. Among older embedding paradigms, we selected F5 [6], $-$F5 [7] which reverses the embedding operation of F5 to increasing the absolute value of DCT coefficients instead of decreasing as in F5, and Jsteg [19]. For modern stego schemes, the payload was fixed at 0.5 bpnzac, while for older schemes it was scaled down to avoid perfect detection by modern steganalyzers. The relative payloads α (in bpnzac) for $-$F5 and Jsteg were set to induce the same number of embedding changes $m = N_{0\text{AC}}H_2^{-1}(\alpha_{-\text{F5}})$ which happens when $\alpha_{\text{Jsteg}} = 2H_2^{-1}(\alpha_{-\text{F5}})$, where H_2 is the binary entropy. The payloads are given in Table I.

A. *LDEAs from the top k insertion test*

Figure 1 shows that for J-MiPOD, a sizable portion of stego images can be detected as stego with only a few inserted 8×8 blocks with stego changes. This is rather surprising because such images have a very small change rate, yet can be detected as stego with high confidence. We say that these images have LDEAs.

	Payload (bpnzac)	P_{E}	MD5	wAUC
J-MiPOD	0.5	.1938	.3837	.9349
J-MiPOD	0.2	.3452	.7033	.8067
J-UNIWARD	0.5	.1967	.4220	.9304
J-UNIWARD	0.2	.3606	.7658	.7792
F5	0.2	.1835	.4292	.9292
−F5	0.05	.0866	.1248	.9827
Jsteg	0.0112	.1315	.2207	.9595

Table I
Detection performance of EfficientNet B4 for stego schemes used in this paper and a mixture of QFs of 75, 90, and 95.

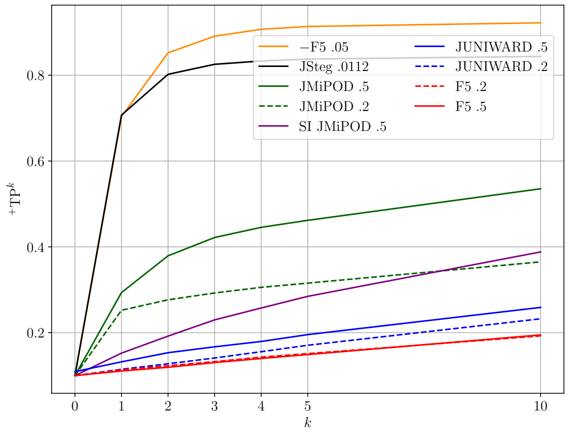


Figure 3. EfficientNet B4's $^{+}\text{TP}^{k}$ rate as a function of top k inserted stego blocks for various stego schemes.

Figure 3 shows the insertion profiles of the same top k insertion test for more embedding schemes with payloads and performance measures shown in Table I. Notice that different embedding schemes have different top k insertion profiles. Also note that the location of such LDEAs in the stego images depends on the actual realization of embedding changes. Different realizations of stego changes for the exact same payload might lead to different LDEAs depending on which DCT coefficients are changed.

The figure also clearly shows that, despite the small payload, Jsteg and –F5 introduce very influential LDEAs as a large percentage of stego images can be identified as stego with only a few blocks with the highest attribution. In contrast, J-UNIWARD and F5 introduce comparatively fewer LDEAs than J-MiPOD. This suggests that for these two algorithms the detector is more an integrator rather than relying on LDEAs.

Conversely, in Figure 4 we show that reverting the changes in the top k blocks and keeping the rest of the stego image intact (i.e. top k canceling instead of insertion) turns the predicted stegos into missed detection. Note that the trends are complementary to those observed for the top k insertion test. The decision threshold was set for 90% True Positive rate.

B. *Do Rich Models catch LDEAs?*

Next, we contrast CNNs and rich models to find out whether rich models can detect LDEAs with any level of confidence.

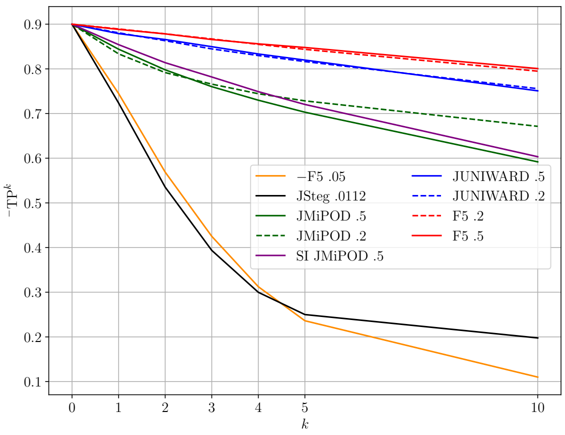


Figure 4. EfficientNet B4's $^{+}\text{TP}^{k}$ rate as a function of top k deleted stego blocks for various stego schemes.

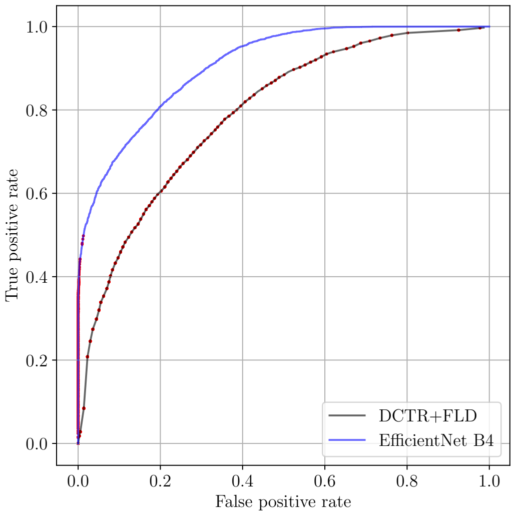


Figure 5. ROC curve of DCTR+FLD ensemble and EfficientNet B4 for J-MiPOD 0.5 bpnzac. Images with strong LDEAs found using IG and EfficientNet B4 are represented by red dots.

To this end, we define the concept of a "strong LDEA" to eliminate cases when the cover image already had a score close to the decision threshold. We consider $k = 1$ and adjust the threshold for $^{+}c^{1}$ to have a 1% FA rate while keeping the thresholds for FAs at 10% as before. For example, for J-MiPOD at 0.5 bpnzac, images with a strong LDEA must have $f(^{+}c^{1}) \geq 0.89$ and $f(c) \leq 0.55$.

Images with strong LDEAs are usually located at the left-most side of the ROC curve for EfficientNet as shown in Figure 5, which shows the images with strong LDEAs as red dots. While they are easy to detect by a CNN even when the only changes made in the image are in one block, in contrast, for DCTR+FLD ensemble, LDEAs are not particularly easy to detect as stego images as shown in Figure 5, where the red dots are scattered rather randomly on the ROC curve. Rich models (DCTR in this case) do not catch LDEAs because of their inability to utilize localized artifacts.

C. *Case study 1: J-MiPOD*

Figure 4 and the previous sections discussed the existence of LDEAs introduced by J-MiPOD, which provide overwhelming evidence to a CNN detector to predict the stego class. Figure 6 shows some examples of LDEAs that are visually identifiable. To further understand the nature of the LDEAs, in Figure 7 we show the average changes of DCT coefficients in each mode computed over test images containing strong LDEAs. It shows that LDEA blocks have (i) a larger change rate than the average 8×8 block of J-MiPOD (ii) more changes in high frequency DCT coefficients. These coefficients are usually zeros in covers, and changing them to ± 1 creates unnatural artifacts. Figure 8 shows that, indeed, the LDEA blocks of J-MiPOD have many more changes in zero coefficients than on average. The distribution for J-UNIWARD is given for reference.

Additionally, LDEAs transfer between different architectures. For J-MiPOD 0.5 bpnzac 82% of SRNet's images with LDEAs are shared with EfficientNet. Reverting the changes in top 3 influential blocks leads to a substantial increase in Missed Detection (from .3883 to .4975 in terms of MD5 as seen in Figure 4) for EfficientNet B4, while the DCTR+FLD ensemble missed detection stays mainly unaffected (from .6963 to .6831 in terms of MD5). Moreover, retraining SRNet on a new dataset where the top 3 influential blocks (computed using B4) have been reverted in all images does not bridge that gap and still produces a significantly worse detector (from .4097 to .4878 in terms of MD5).

D. *Case study 2: J-UNIWARD*

Figure 4 shows that J-UNIWARD introduces significantly fewer LDEAs than J-MiPOD even though their detectability is very similar (Table I). In fact, Figure 2 already shows an interesting difference between the two embedding schemes when looking at their last activation map: J-UNIWARD images tend to activate the majority of the map, whereas J-MiPOD images activate a highly localized area. The ranges of IG block attributions also differ with J-UNIWARD exhibiting a rather spatially uniform attribution map unlike J-MiPOD. This seems to indicate that the network is an "integrator" for most J-UNIWARD images, while it also utilizes localized information for J-MiPOD.

To confirm this conjecture, for each image we count the number of elements in the last activation map that exceed a threshold set as $3\times$ the average of the last activation map (as we try to identify "large logit cells" or spikes in the map). Figure 9 shows the histograms of these counts across $6,000$ test images. For J-UNIWARD, these large logit cells are almost non-existent, while for J-MiPOD and especially J-MiPOD images with strong LDEA blocks many last activation maps are comprised of such spikes.

E. *Case study 3: Jsteg*

Figure 4 shows that Jsteg introduces many LDEAs, which is not surprising since Jsteg is not content adaptive and highly likely to produce detectable artifacts. On average, the top 1

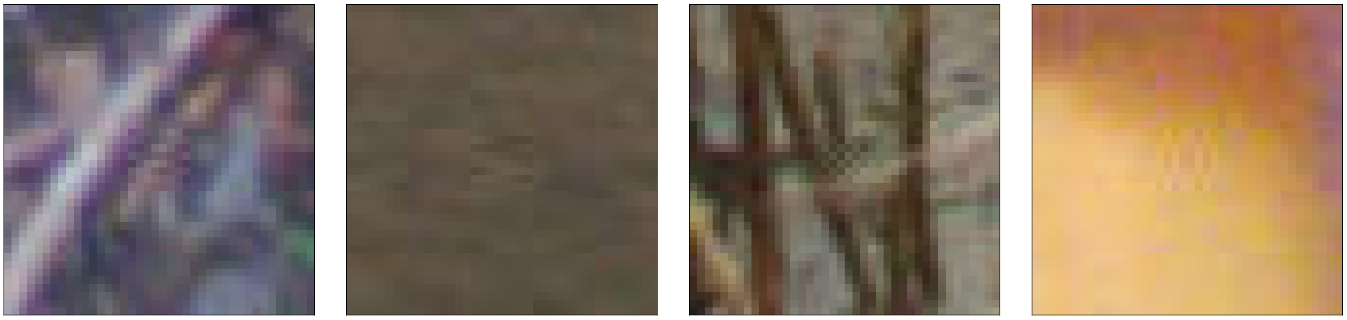
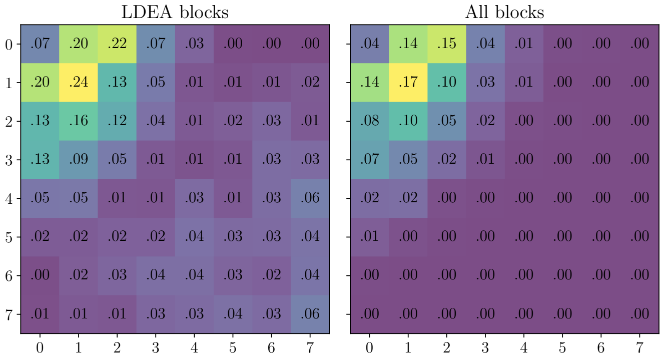


Figure 6. Example of visible local traces of J-MiPOD. The center 8 × 8 block is the top 1 influential block using IG. Left to right images: '05626.jpg', '47211.jpg', '48020.jpg', and '55961.jpg'.



LDEA blocks

	0	1	2	3	4	5	6	7
0	.07	.20	.22	.07	.03	.00	.00	.00
1	.20	.24	.13	.05	.01	.01	.01	.02
2	.13	.16	.12	.04	.01	.02	.03	.01
3	.13	.09	.05	.01	.01	.01	.03	.03
4	.05	.05	.01	.01	.03	.01	.03	.06
5	.02	.02	.02	.02	.04	.03	.03	.04
6	.00	.02	.03	.04	.04	.03	.02	.04
7	.01	.01	.01	.03	.03	.04	.03	.06

All blocks

	0	1	2	3	4	5	6	7
0	.04	.14	.15	.04	.01	.00	.00	.00
1	.14	.17	.10	.03	.01	.00	.00	.00
2	.08	.10	.05	.02	.00	.00	.00	.00
3	.07	.05	.02	.01	.00	.00	.00	.00
4	.02	.02	.00	.00	.00	.00	.00	.00
5	.01	.00	.00	.00	.00	.00	.00	.00
6	.00	.00	.00	.00	.00	.00	.00	.00
7	.00	.00	.00	.00	.00	.00	.00	.00

Figure 7. Average changes per mode for LDEA blocks (left) and over all blocks (right) computed over test images containing strong LDEAs for J-MiPOD.

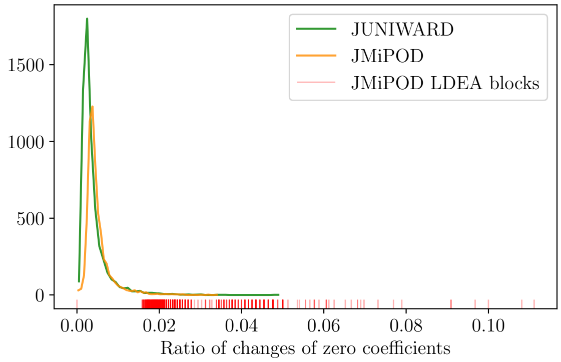


Figure 8. Histogram of the ratio of changes on zero coefficients w.r.t. all changes for J-UNIWARD and J-MiPOD. Rug plot data points correspond to the same ratio computed only on strong LDEA 8 × 8 blocks of J-MiPOD.

influential blocks of Jsteg have 98.01% of changes *increasing* the absolute value of the DCT coefficients, whereas on average across all blocks Jsteg increases the absolute value of DCT coefficients with a rate of only 65.06%. Increasing the absolute value increases the block variance, which makes it easier to detect in a smooth area.

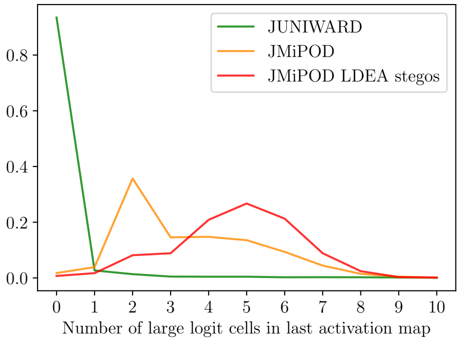


Figure 9. Normalized histogram of the number of large logit cells of the last activation map of EfficientNet B4 for J-UNIWARD, J-MiPOD, and J-MiPOD with strong LDEAs.

F. Case study 4: F5, –F5

Figure 4 shows that F5 introduces very few LDEAs. Unlike other schemes, F5 only decreases the absolute value of DCT coefficients. For $-$F5, the LDEAs count is the largest. The culprit is the embedding operation of increasing the absolute value of DCT coefficients as it adds artificial content to 8 × 8 DCT blocks. This is further confirmed by comparing –F5 with Jsteg with payload scaled to have the same number of changes as $-$F5. While Jsteg's curve is lower than $-$F5 for $k > 1$, both have the same number of strong LDEAs (for $k = 1$).

V. Multi-class detectors and stego inhibition

In this section, we briefly study a multi-class CNN detector that uses inhibition to distinguish between embedding algorithms. To this end, we purposely selected two embedding algorithms with very different selection channels: J-UNIWARD and bUERD, which is a version of UERD that was used in the ALASKA II competition. An implementation mistake made bUERD's selection channel anomalous with the embedding changes concentrated around the image boundary in most stego images. A network able to see LDEAs should discover this flaw and exploit it for detection.

Our multi-class detector was the EfficientNet B4 trained using multi-class cross-entropy loss. In this section, we drop the modulation by $s - b$ in Eq. 1 since we are interested in

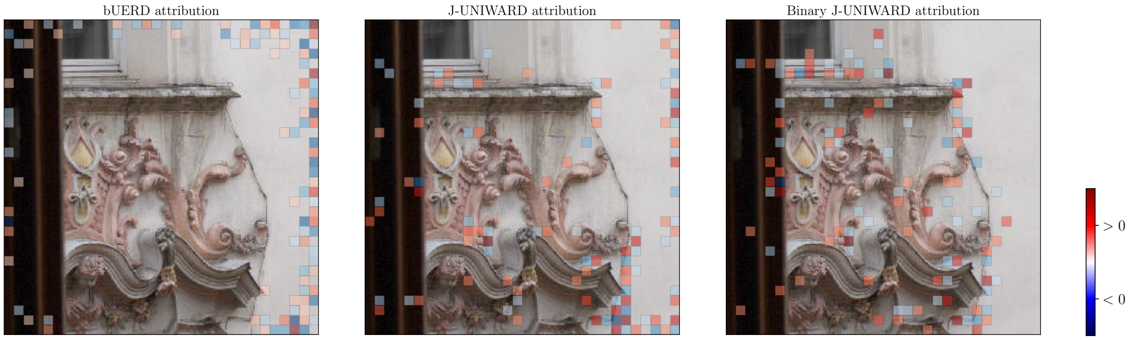


Figure 10. Attribution maps for image '43201.jpg' from the ALASKA II dataset embedded with J-UNIWARD and bUERD from the multi-class, and binary J-UNIWARD EfficientNet B4 (from left to right). Notice the anti-correlated attributions in the right boundary of both images from the multi class attributions, which are not visible in the binary attributions. The figure shows only the 90% largest attributions for each map in absolute value for visual clarity.

blocks with changes by both bUERD and J-UNIWARD not only a by one of them at a time.

Given a J-UNIWARD image, the stego attribution is typically high in blocks with complex content, while an inhibitory attribution at the image boundary. For the bUERD version of the same image, the attributions at the image boundary are often anti-correlated with the attributions from the J-UNIWARD image. We call this phenomenon "stego inhibition" as the CNN uses artifact traces from all stego schemes it was trained on. Another intuitive explanation of stego inhibition is when the CNN predicts "this is a J-UNIWARD image and not bUERD" using inhibition of bUERD artifacts. An example of this is shown in Figure 10. We compare this to a known phenomenon in computer vision and neurology where neurons explicitly inhibit against features that do not make sense in certain spatial areas. In this case, the stego noise at the image boundary is representative of bUERD, and does not make sense for images other than bUERD (provided the boundary does not contain complex content). Also notice in Figure 10 that the attribution of the J-UNIWARD image from both the binary and multi-class detectors have strong similarities (outside the right boundary), which means that both detectors have converged to detecting similar patterns.

VI. Conclusions

Using attribution tools, we provide evidence for the popular belief that CNNs reach their decision by detecting local embedding artifacts. By analyzing modern content-adaptive schemes and older embedding paradigms, we characterize these artifacts and show that they are mostly associated with high frequency content-creating changes. CNNs ability to leverage localized signals plays a role in distinguishing between selection channels of different embedding algorithms when training a multi-class detector.

This work was supported by NSF grant No. 2028119.

References

[1] M. Chaumont, "Deep learning in steganography and steganalysis," in *Digital Media Steganography: Principles, Algorithms, Advances*, ch. 14, pp. 321–349, 2020.

[2] R. Cogranne and J. Fridrich, "Modeling and extending the ensemble classifier for steganalysis of digital images using hypothesis testing theory," *IEEE TIFS*, vol. 10, pp. 2627–2642, December 2015.

[3] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE TIFS*, vol. 7, pp. 432–444, April 2012.

[4] R. Cogranne, Q. Giboulot, and P. Bas, "Steganography by minimizing statistical detectability: The cases of JPEG and color images," in *Proc. of the 8th ACM IH&MMSEC*, 2020.

[5] 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 MMSEC Workshop*, vol. 2014:1, 2014.

[6] J. Fridrich, T. Pevný, and J. Kodovský, "Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities," in *Proc. of the 9th ACM MMSEC*, (Dallas, TX), pp. 3–14, September 20–21, 2007.

[7] T. Pevný and J. Fridrich, "Novelty detection in blind steganalysis," in *Proc. of the 10th ACM MMSEC*, (Oxford, UK), pp. 167–176, September 22–23, 2008.

[8] R. Cogranne, Q. Giboulot, and P. Bas, "ALASKA#2: Challenging academic research on steganalysis with realistic images," in *Proc. of the 12th IEEE WIFS*, (Held virtually), December 6–11, 2020.

[9] T. Mingxing and V. L. Quoc, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. of the 36th ICML*, vol. 97, pp. 6105–6114, June 9–15, 2019.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. of the 2009 IEEE/CVF CVPR*, pp. 248–255, June 20–25, 2009.

[11] Y. Yousfi, J. Butora, E. Khvedchenya, and J. Fridrich, "ImageNet pretrained CNNs for JPEG steganalysis," in *The 12th IEEE WIFS*, (Held virtually), December 6–11, 2020.

[12] Y. Yousfi, J. Butora, J. Fridrich, and C. Fuji Tsang, "Improving efficientnet for JPEG steganalysis," in *Proc. of the 9th ACM IH&MMSEC*, June 22–25, 2021.

[13] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE TIFS*, vol. 14, pp. 1181–1193, May 2019.

[14] V. Holub and J. Fridrich, "Low-complexity features for JPEG steganalysis using undecimated DCT," *IEEE TIFS*, vol. 10, pp. 219–228, February 2015.

[15] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. of the 34th ICML*, pp. 3319–3328, 2017.

[16] E. J. Friedman, "Paths and consistency in additive cost sharing," *International Journal of Game Theory*, vol. 32, no. 4, pp. 501–518, 2004.

[17] P. Sturmfels, S. Lundberg, and S.-I. Lee, "Visualizing the impact of feature attribution baselines," *Distill*, 2020. http://distill.pub/2020/attribution-baselines.

[18] L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient JPEG steganography," *IEEE TIFS*, vol. 9, pp. 814–825, May 2014.

[19] D. Upham, "Steganographic algorithm JSteg." Software available at http://zooid.org/~paul/crypto/jsteg.