Adversarial Images through Stega Glasses

Benoît Bonnet, Teddy Furon, Patrick Bas

Adversarial Images through Stega Glasses

Benoît Bonnet
Univ. Rennes, Inria, CNRS, IRISA
Rennes, France
benoit.bonnet@inria.fr

Teddy Furon
Univ. Rennes, Inria, CNRS, IRISA
Rennes, France
teddy.furon@inria.fr

Patrick Bas
Univ. Lille, CNRS, Centrale Lille, UMR 9189, CRIStAL, Lille, France
patrick.bas@centralelille.fr

***Abstract*—This paper explores the connection between steganography and adversarial images. On the one hand, steganalysis helps in detecting adversarial perturbations. On the other hand, steganography helps in forging adversarial perturbations that are not only invisible to the human eye but also statistically undetectable. This work explains how to use these information hiding tools for attacking or defending computer vision image classification. We play this cat and mouse game with state-of-art classifiers, steganalyzers, and steganographic embedding schemes. It turns out that steganography helps more the attacker than the defender.**

I. Introduction

Adversarial samples is an emerging field in Information Forensics and Security, addressing the vulnerabilities of Machine Learning algorithms. This paper casts this topic to the application of Computer Vision, and in particular, image classification. A Deep Neural Network is trained to classify images depending on the type of object represented in the picture. This is for instance the well-known ImageNet challenge encompassing a thousand of classes. The state-of-the-art proposes impressive results as classifiers do a better job than humans with less classification errors and much faster timings. This may deserve the wording 'Artificial Intelligence' as a computer now compete with humans on a difficult task.

The literature of adversarial samples reveals that these classifiers are vulnerable to specific image modifications. For a given image, an attacker can craft a perturbation that triggers a wrong classification. This perturbation is often a weak signal barely visible to the human eyes. Almost surely, no human would incorrectly classify these adversarial images. This topic is extremely interesting as it challenges the 'Artificial Intelligence' qualification too soon attributed to Deep Learning.

A. Defenses

We can find in the literature four types of defenses or counter-attacks to deal with adversarial contents:
To detect: Being barely visible does not mean that the perturbation is not statistically detectable. This defense analyses the image and bypasses the classifier if detected as adversarial [1].
To reform: The perturbation looks like a random noise that may be filtered out. This defense is usually a front-end projecting the image back to the manifold of natural images [2].
To robustify: At learning, adversarial images are included in the training set with their original class labels. Adversarial retraining usually robustifies a 'naturally' trained network [3].
To randomize: At testing, the classifier depends on a secret key or an alea. This blocks pure white-box attacks [4], [5].

This paper deals with the first line of defense. It is a pity that most papers proposing a defense do not seriously challenge it. Security is often overclaimed as shown in [6], [7].

B. Connections with Information Hiding

Paper [8] makes the connection between Adversarial Samples and Information Hiding (be it watermarking or steganography). Both fields modify images (or any other type of media) in the spatial domain so that the content is moved to a targeted region of the feature space. That region is the region associated to a secret message in Information Hiding or to a wrong class in Adversarial Sampling. Indeed, paper [8] shows that Adversarial Sampling benefits from ideas proven efficient in Watermarking, and vice-versa.

This paper contributes to the same spirit by investigating what both Steganography and Steganalysis bring to Adversarial Sampling. There are two natural ideas:
Steganalysis is the art of detecting weak perturbations in images. This field is certainly useful for the defender.
Steganography is the art of modifying an image while being non-detectable. This field is certainly useful for the attacker.

These two sides of the same coin allow to mount a defense and to challenge it in return. This paper aims at revealing the status of the game between the attacker and the defender at the time of writing, *i.e.* when both players use up-to-date tools: state-of-the-art image classifiers with premium steganalyzers, and best-in-class steganography embedders. As far as we know, this paper proposes three first time contributions:

- Assess the robustness of very recent image classifiers, EfficientNet [9] and its robust version [10],
- Apply the best steganalyzer SRNET [11] to detect adversarial images,
- Use the best steganographic schemes to craft perturbations: HILL [12] uses empirical costs, MiPod [13] models undetectability from a statistical point of view, while GINA [14], [15] synchronizes embeddings on color channels.

WIFS'2020, December, 6-9, 2020, New York, USA. XXX-X-XXXX-XXXX-X/XX/$XX.00

II. State of the art

A. Steganalysis is Versatile

Steganalysis has always been bounded to steganography, obviously. Yet, a recent trend is to resort to this tool for other purposes than detecting whether an image conceals a secret message. For instance, paper [16] claims the universality of SRM and LBP steganalyzers to detect image processing (like Gaussian blurring, gamma correction) and splicing. The authors of [17] used this approach during the IEEE IFS-TC image forensics challenge. The same trend holds as well on audio forensics [18]. As for camera model identification, the inspiration from steganalysis (co-occurrences, color dependencies, conditional probabilities) is clearly apparent in [19].

This reveals a certain versatility in steganalysis. It is not surprising since the main goal is to model and detect weak signals. Modern steganalyzers are no longer based on hand-crafted features like SRM [20]. They are no more no less than Deep Neural Networks like XU-Net [21] or SRNET [11]. The frontier between steganalysis and any two-class image classification problem (such as image manipulation detection) is blurred. Yet, these networks have a specific structure able to focus on weak signal detection. They for example avoid pooling operations in order to preserve high frequency signals, they also need large databases combined with augmentation techniques and curriculum learning to converge [22].

This general-purpose based on steganalysis method has some drawbacks. It lacks fine-grained tampering localization, which is an issue in forensics [23]. Paper [24] goes a step further in the cat-and-mouse game with an anti-forensic method: knowing that the defender uses a steganalyzer, the attacker modifies the perturbation (accounting for a median filtering or a contrast enhancement) to become less detectable.

As for adversarial images detection, this method is not new as well. The authors of [25] wisely see steganalysis detection as a perfect companion to adversarial re-training. This last mechanism fights well against small perturbation. It however struggles in correctly classifying coarser and more detectable attacks. Unfortunately, this idea is supported with a proof of concept (as acknowledged by the authors): the steganalyzer is rudimentary, the dataset is composed of tiny images (MNIST). On the contrary, the authors of [26] outline that steganalysis works better on larger images like ImageNet (ILSVRC-2016). They however use a deprecated classifier (VGG-16) with outdated steganalyzers based on hand-crafted features (SPAM and SRM).

Conversely, adversarial samples recently became a source of inspiration for steganography: paper [27] proposes the concept of steganography with an adversarial embedding fooling a DNN-based steganalyzer.

B. Adversarial Images

This paper focuses on white-box attacks where the attacker knows all implementation details of the classifier.

To make things clearer, the classifier has the following structure: a pre-processing T maps an image $\mathbf{I}_o \in \{0, 1, \dots, 255\}^n$ (with $n = 3LC$, 3 color channels, L lines and C columns of pixels) to $\mathbf{x}_o = \mathsf{T}(\mathbf{I}_o) \in \mathcal{X}^n$, with $\mathcal{X} := [0, 1]$ (some networks also use $\mathcal{X} = [-1, 1]$ or $[-3, 3]$). This pre-processing is heuristic, sometimes it just divides the pixel value by 255, sometimes this normalization is channel dependent based on some statistics (empirical mean and standard deviation). This flattened vector $\mathbf{x}_o$ is fed the trained neural network to produce the estimated probabilities $(\hat{p}_k(\mathbf{x}_o))_k$ of being from class $k \in \{1, \dots, K\}$. The predicted class is given by:

$$\hat{c}(\mathbf{x}_o) = \arg\max_k \hat{p}_k(\mathbf{x}_o). \tag{1}$$

The classification is correct if $\hat{c}(\mathbf{x}_o) = c(\mathbf{x}_o)$, the ground truth label of image I_o.

An *untargeted* adversarial attack aims at finding the optimal point:

$$\mathbf{x}_a^\star = \arg \min_{\mathbf{x}:\hat{c}(\mathbf{x}) \neq c(x_o)} \|\mathbf{x} - \mathbf{x}_o\|, \tag{2}$$

where $\| \cdot \|$ is usually the Euclidean distance.

Discovering this optimal point is difficult because the space dimension n is large. In a white-box scenario, all attacks are sub-optimal iterative processes. They use the gradient of the network function efficiently computed thanks to the back-propagation mechanism to find a solution $\mathbf{x}_a$ close to $\mathbf{x}_a^\star$. They are compared in terms of probability of success, average distortion, and complexity (number of gradient computations). This paper considers well-known attacks: FGSM [28], PGD (Euclidean version) [3], DDN [29], and CW [30] (ranked from low to high complexity).

As outlined in [31], definition (2) is very common in literature, yet it is incorrect. The goal of the attacker is to create an adversarial image $\mathbf{I}_a$ in the pixel domain. Applying the inverse mapping T^{-1} is not solving the issue because this a priori makes non integer pixel values. Rounding to the nearest integer, $\mathbf{I}_a = \lceil \mathsf{T}^{-1}(\mathbf{x}_a) \rfloor$, is simple but not effective. Some networks are so vulnerable (like ResNet-18) that $\mathsf{T}^{-1}(\mathbf{x}_a) - \mathbf{I}_o$ is a weak signal partially destroyed by rounding. The impact is that, after rounding, $\mathbf{I}_a$ is no longer adversarial. Note that DDN is a rare example of a powerful attack natively offering quantized pixel values.

Paper [31] proposes a post-processing Q on top of any attack that makes sure $\mathbf{I}_q = \mathsf{Q}(\mathsf{T}^{-1}(\mathbf{x}_a))$ is *i)* an image (integral constraint), *ii)* remains adversarial, and *iii)* has a low Euclidean distortion $\|I_q - I_o\|$. This paper follows the same approach but adds another constraint: *iv)* be non-detectable.

C. Steganographic Embeddings

Undetectability is usually tackled by the concept of costs in the steganographic literature: each pixel location i of the cover image is assigned a set of costs $(w_i(\ell))_\ell$ that reflects the detectability of modifying the i-th pixel by ℓ quantum. Usually, $w_i(0) = 0$, $w_i(-\ell) = w_i(\ell)$, and $w_i(|\ell|)$ is increasing. The goal of the steganographer is to embed a message $\mathbf{m}$ while minimizing the empirical steganographic distortion:

$$D(\boldsymbol{\ell}) := \sum_{i=1}^{n} w_i(\ell_i). \tag{3}$$

This is practically achieved using Syndrome Trellis Codes [32]. Note that this distortion is additive, which is equivalent to consider that each modification yields to a detectability which is independent from the others.

We propose to use the steganographic distortion (instead of L_1, L_2 or L_∞ norms in adversarial literature) in order to decrease detectability. There are strategies to take into account potential interactions between neighboring modifications. The image can first be decomposed into disjoint lattices to be sequentially embedded. And costs can then be sequentially updated after the embedding of every lattice [14]. This work uses three different families of steganographic costs.

The first one, HILL [12], is empirical and naive, but has nevertheless been widely used in steganography and is easy to implement. The cost map $\mathbf{w}$ associated to ± 1 is computed using 2 low-pass averaging filters $\mathbf{L}_1$ et $\mathbf{L}_2$ of respective size 3×3 et 15×15 and one high pass filter $\mathbf{H}$:

$$\mathbf{w} = \frac{1}{|\mathbf{I} * \mathbf{H}| * \mathbf{L}_1} * \mathbf{L}_2, \text{with } \mathbf{H} = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}. \quad (4)$$

The second one, derived from MiPod [13], assumes that the residual signal is distributed as $\mathcal{N}(0, \sigma_i^2)$ for the original image, and $\mathcal{N}(\ell_i, \sigma_i^2)$ for the stego image. The variance σ_i^2 is estimated on each pixel using Wiener filtering and a least square approximation on a basis of cosine functions. The cost is the log likelihood ratio between the two distributions evaluated at 0, i.e.:

$$w_i(\ell_i) = \ell_i^2 / \sigma_i^2. \quad (5)$$

Unlike the previous one, this model can handle modifications other than ± 1.

The last one is a cost updating strategy favoring coherent modifications between pixels within a spatial or color neighborhood. It is called GINA [15] is derived from CMD [14]. It splits the color images into 4 disjoint lattices per channel, i.e. 12 lattices. The embedding performs sequentially starting by the green channel lattices. The costs on one lattice is updated according to the modifications done on the previous ones as:

$$w_i'(\ell_i) = \frac{1}{9} w_i(\ell_i), \text{ if } \mathsf{sign}(\ell_i) = \mathsf{sign}(\mu_i), \quad (6)$$

with μ_i the average of the modifications already performed in the neighborhood of location i.

III. Steganographic Post-Processing

This section presents the use of steganography in our post-processing Q mounted on top of any adversarial attack.

A. About Steganography and Adversarial Examples

Paper [25] stresses a fundamental difference: Steganalysis has two classes, where the class 'cover' distribution is given by Nature, whereas the class 'stego' distribution is a consequence of designed embedding schemes. On the other hand, a *perfect* adversarial example and an original image are distributed as by the class $\hat{c}(\mathbf{x}_a)$ or $c(\mathbf{x}_o)$, which are both given by Nature.

We stress another major difference: Steganographic embedding is essentially a stochastic process. Two stego-contents derived from the same cover are different almost surely. This is a mean to encompass the randomness of the messages to be embedded. This is also the reason why steganographic embedders turns the costs $(w_i(\ell))_\ell$ into probabilities $(\pi_i(\ell))_\ell$ of modifying the i-th pixel by ℓ quantum. These probabilities are derived to minimize the detectability under the constraint of an embedding rate given by the source coding theorem:

$$R = -n^{-1} \sum_i \sum_{\ell_i} \pi_i(\ell_i) \log_2 (\pi_i(\ell_i)) \text{ bits}. \quad (7)$$

In contrast, an attack is a deterministic process always giving the same adversarial version of one original image. Adversarial imaging does not need these probabilities.

B. Optimal post-processing

Starting from an original image, we assume that an attack has produced $\mathbf{x}_a$ mapped back to $\mathbf{I}_a = \mathsf{T}^{-1}(\mathbf{x}_a)$. The problem is that $\mathbf{I}_a \in [0, 255]^n$, *i.e.* its pixel values are a priori not quantized. Our post-processing specifically deals with that matter, outputting $\mathbf{I}_q = \mathsf{Q}(\mathbf{I}_a) \in \{0, \dots, 255\}^n$. We introduce $\mathbf{p}$ the perturbation after the attack and $\mathbf{q}$ the perturbation after our post-processing:

$$\mathbf{p} := \mathbf{I}_a - \mathbf{I}_o \in \mathbb{R}^n, \quad (8)$$

$$\boldsymbol{\ell} := \mathbf{I}_q - \mathbf{I}_o \in \mathbb{Z}^n. \quad (9)$$

The design of Q amounts to find a good $\boldsymbol{\ell}$. This is more complex than just rounding perturbation $\mathbf{p}$.

We first restrict the range of $\boldsymbol{\ell}$. We define the degree of freedom d as the number of possible values for each ℓ_i, $1 \leq i \leq n$. This is an even integer greater or equal than 2. The range of ℓ_i is centered around p_i. For instance, when $d = 2$, $\ell_i \in \{\lfloor p_i \rfloor, \lceil p_i \rceil\}$. In general, the range is given by

$$\mathcal{L}_i := \{\lceil p_i \rceil - d/2, \dots, \lceil p_i \rceil - 1, \lceil p_i \rceil, \dots, \lceil p_i \rceil + d/2 - 1\}. \quad (10)$$

Over the whole image, there are d^n possible sequences for $\boldsymbol{\ell}$.

We now define two quantities depending on $\boldsymbol{\ell}$. The *classifier loss* at $\mathbf{I}_q = \mathbf{I}_a - \mathbf{p} + \boldsymbol{\ell}$:

$$L(\boldsymbol{\ell}) := \log(\hat{p}_{c_o}(\mathbf{I}_a - \mathbf{p} + \boldsymbol{\ell})) - \log(\hat{p}_{c_a}(\mathbf{I}_a - \mathbf{p} + \boldsymbol{\ell})), \quad (11)$$

where c_o is the ground truth class of $\mathbf{I}_o$ and c_a is the predicted class after the attack. When the attack succeeds, it means that $\mathbf{I}_a$ is classified as $c_a \neq c_o$ because $\hat{p}_{c_a}(\mathbf{I}_a) > \hat{p}_{c_o}(\mathbf{I}_a)$ so that $L(\mathbf{p}) < 0$. Our post-processing cares about maintaining this adversariality. This constrains $\boldsymbol{\ell}$ s.t. $L(\boldsymbol{\ell}) < 0$.

The second quantity is the *detectability*. We assume that a black-box algorithm gives the stego-costs $(w_i(\ell))_\ell$ for a given original image. The overall detectability of $\mathbf{I}_q$ is gauged by $D(\boldsymbol{\ell})$ (3). In the end, the optimal post-processing Q minimizes detectability while maintaining adversariality:

$$\boldsymbol{\ell}^\star = \arg \min_{\boldsymbol{\ell}: L(\boldsymbol{\ell}) < 0} D(\boldsymbol{\ell}). \quad (12)$$

C. Our proposal

The complexity for finding the solution of (12) a priori scales as $O(d^n)$. Two ideas from the adversarial examples literature help reducing this. First, the problem is stated as an Lagrangian formulation as in [30]:

$$\boldsymbol{\ell}_\lambda = \arg\min D(\boldsymbol{\ell}) + \lambda L(\boldsymbol{\ell}). \tag{13}$$

where $\lambda \geq 0$ is the Lagrangian multiplier. This means that we must solve this problem for any λ and then find the smallest value of λ s.t. $L(\boldsymbol{\ell}_\lambda) < 0$.

Second, the classifier loss is linearized around $\mathbf{I}_a$, *i.e.* for $\boldsymbol{\ell}$ around $\mathbf{p}$: $L(\boldsymbol{\ell}) \approx L(\mathbf{p}) + (\boldsymbol{\ell} - \mathbf{p})^\top \mathbf{g}$, where $\mathbf{g} = \nabla L(\mathbf{p})$. This transforms problem (13) into

$$\boldsymbol{\ell}_\lambda = \arg\min \sum_{i=1}^{n} w_i(\ell_i) + \lambda(p_i - \ell_i).g_i. \tag{14}$$

The solution is now tractable because the functional is separable: we can solve the problem pixel-wise. The algorithm stores in $d \times n$ matrix W the costs, and in $d \times n$ matrix G the values $((p_i - \ell_i).g_i)_i$ for $\ell_i \in \mathcal{L}_i$ (10). For a given λ, it computes $W + \lambda G$ and looks for the minimum of each column $1 \leq i \leq n$. In other words, it is as complex as n minimum findings, each over d values, which scales as $O(n \log d)$.

Note that for $\lambda = 0$, Q quantizes $I_{a,i}$ 'towards' $I_{o,i}$ to minimize detectability. Indeed, if $\ell_i = 0$ is admissible ($0 \in \mathcal{L}_i$ holds if $|p_i| \leq d/2$), then $\mathsf{Q}(I_{a,i}) = I_{o,i}$ at $\lambda = 0$.

On top of solving (14), a line search over λ is required. The linearization of the loss being a crude approximation, we make calls to the network to check that $\mathsf{Q}(\mathbf{I}_a)$ is adversarial: When testing a given value of λ, $\boldsymbol{\ell}_\lambda$ is computed to produce I_q that feeds the classifier. If I_q is adversarial then $L(\boldsymbol{\ell}_\lambda) < 0$ and we test a lower value of λ (giving more importance to the detectability), otherwise we increase it. We use a binary search with a stopping criterion to control complexity of the post-processing. The search stops when two successive values of λ are different by less than 1,000. Optimal λ varies widely between different images. This criterion was empirically set to give both optimal value and short research time.

D. Simplification for quadratic stego-costs

We now assume that the stego-costs obey to the following expression: $w_i(\ell) = \ell^2/\sigma_i^2$. This makes the functional of (14) (restricted to the i-th pixel) equals to $\ell_i^2/\sigma_i^2 - \lambda g_i \ell_i + \lambda p_i$ which minimizer is $\tilde{\ell}_i = \lambda g_i \sigma_i^2/2$.

Yet, this value a priori does not belong to $\mathcal{L}_i$ (10). This is easily solved because a quadratic function is symmetric around its minimum, therefore the minimum over $\mathcal{L}_i$ is its value closest to $\tilde{\ell}_i$ as shown in Fig. 1. The range $\mathcal{L}_i$ being nothing more than a set of consecutive integers, we obtain a closed form expression:

$$\ell_{\lambda,i} = \min(\max([\lambda g_i \sigma_i^2/2], \lceil p_i \rceil - d/2), \lceil p_i \rceil + d/2 - 1), \tag{15}$$

where $[\cdot]$ is the rounding to the nearest integer. The post-processing has now a linear complexity.

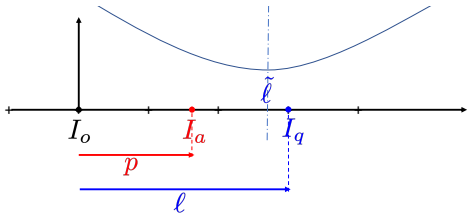

Fig. 1. Rounding the minimizer when the stego-cost is quadratic.

In this equation, the min and max operate a clipping so that $\ell_{\lambda,i}$ belongs to $\mathcal{L}_i$. This clipping is active if $\tilde{\ell}_i \notin \mathcal{L}_i$, which happens if $\lambda \geq \bar{\lambda}_i$ with

$$\bar{\lambda}_i := \begin{cases} \left| \frac{2\lceil p_i \rceil - d}{g_i \sigma_i^2} \right|_+ & \text{if } g_i < 0 \\ \left| \frac{2\lceil p_i \rceil + d - 2}{g_i \sigma_i^2} \right|_+ & \text{if } g_i > 0, \end{cases} \tag{16}$$

where $|a|_+ = a$ if $a > 0$, 0 otherwise. This remark is important because it shows that for any $\lambda > \max_i \bar{\lambda}_i$, the solution $\boldsymbol{\ell}_\lambda$ of (15) remains the same due to clipping. Therefore, we can narrow down the line search of λ to $[0, \max_i \bar{\lambda}_i]$.

IV. Experimental Investigation

A. Experimental setup

Our experimental work uses 18,000 images from ImageNet of dimension 224×224×3. This subset is split in 1,000 for testing and comparing, 17,000 for training. An image is attacked only if the classifier predicts its correct label beforehand. This happens with probability equaling the accuracy of the network Acc. We measure $\overline{L_2}$ the average Euclidean distance of the perturbation $\boldsymbol{\ell}$ and P_{suc} the probability of a successful attack *only over correctly labeled images*.

We attack the networks with 4 different attacks: FGSM [28], PGD_2 [3], CW [30] and DDN [29]. All of the attacks are run in a *best-effort* fashion with a complexity limited to 200 iterations. For FGSM and PGD_2 the distortion is gradually increased until the image is adversarial. For more complex CW and DDN, different sets of parameters are used on a total maximum of 200 iterations. The final attacked version is the adversarial image with the smaller distortion. DDN is the only attack that creates integer images. The other 3 are post-processed either by the enhanced quantization [31], which is our baseline, or by our method explained in Sect. III-C.

The adversarial image detectors are evaluated by the true positive rate TPR when the false positive rate is fixed to 5%.

B. Robustness of recent classifiers: there is free lunch

Our first experiment compares the robustness of the famous ResNet-50 network to the recent classifiers: the natural version of EfficientNet-b0 [9] (Nat) and its robust version trained with AdvProp [10] (Rob). Note that the authors of [10] apply adversarial re-training for improving accuracy. As far as we known, the robustness of this version is not yet established.

Table I confirms that modern classifiers are more accurate and more robust (lower P_{suc} and/or bigger L_2). This is indeed

TABLE I
ROBUSTNESS OF RECENT CLASSIFIERS AGAINST PGD_2 FOLLOWED BY QUANTIZATION [31]

	Acc (%)	P_{suc} (%)	$\overline{L_2}$
ResNet-50	80.0	97.2	81
'Nat' EfficientNet-b0 [9]	82.8	88.0	88
'Rob' EfficientNet [10]	84.3	71.8	112

a surprise: It pulls down the myth of 'No Free Lunch' in adversarial machine learning literature [33], [34] (The price to pay for robustifying a network is pretendedly a lower accuracy).

C. Detection with a Steganalyzer

We use three steganalyzers to detect adversarial images. Their training set is composed of 15,651 pairs of original and adversarial images. The latter are crafted with *best-effort* FGSM against natural EfficientNet-b0.

The first detector is trained on SRM feature vectors [20], with dimensions 34,671. SRM is a model that applies to only one channel. It is computed on the luminance of the image in our experimental work. The classifier used to fit these high-dimensional vectors into two classes is the linear regularized classifier [35]. The second detector is based on the color version of SRM: SCRMQ1 [36] with dimension 18,157. The classifier is the same. The third detector is SRNet [11], one of the best detectors in steganalysis. Training is performed on 180 epochs: The first 100 with a learning rate of 10^{-3}, the remaining 80 with 10^{-4}. Data augmentation is also performed during training. First, there is a probability $p_1 = 0.5$ of mirroring the pair of images. Then, there is another probability $p_2 = 0.5$ of rotating them 90 degrees.

The attacks: Table II shows that the probabilities of success P_{suc} are similar except for DDN (a larger complexity increases P_{suc} but it is not the aim of this study). Note that PGD_2 and CW whose samples are quantized with [31] are attacks as reliable as FGSM but with a third of the distortion.

The detectors: Table II gives also the TPR associated to the detectors. Although [26] achieve good performances with SRM, we were not able to reproduce their results. This could be due either to finer attacks or to the effect of quantization. Our results show that the detectors generalize well: although trained to detect images highly distorted by FGSM, they can detect as well and sometimes even better more subtle attacks like CW. Moreover, SRNet always outperforms SCRMQ1 and delivers an impressive accuracy. Table II shows that PGD_2+[31] is the worst-case scenario for defense. The probability of fooling both the classifier EfficientNet-b0 and the detector SRNet combines to only 5.9%.

TABLE II
DETECTION OF ADVERSARIAL IMAGES WITH STEGANALYZERS

	P_{suc}	$\overline{L_2}$	SRM(%)	SCRMQ1(%)	SRNet(%)
FGSM+[31]	89.7	286	72.00	83.3	**93.5**
PGD_2+[31]	88.0	84	65.02	81.2	**93.3**
CW+[31]	89.7	97	68.78	83.6	**94.5**
DDN	83.2	186	79.53	91.9	**94.8**

'Angora rabbit' 'woolen'

'hare', $L_2 = 488$ 'knot', $L_2 = 449$

Fig. 2. Top row: Cover images with their label below. Bottom row: adversarial images with steganographic embedding GINA (d=4). Below them are their new label and the distortion

D. Post-processing with a Steganographic Embedder

We now play the role of the attacker. We use PGD_2 with best effort as the base attack to compare the detectability of four post-processings: The non-steganographic insertion [31] as a baseline, HILL (4), MiPod (5), and GINA (6). GINA uses the quadratic method explained in Sect. III-D sequentially over the 12 lattices. Quadratic stego-costs are updated with CMD strategy (6). Each lattice contributes to a 1/12 of the initial classification loss.

Distortion increases with each method and along the degree of freedom d. Steganographic embedding therefore reduces detectability at the cost of increased distortion. From the attacker perspective, the best-case scenario with PGD_2 is with GINA at d=2 as seen on Table III. This scenario now has 69.9% chance of fooling both the classifier and the detector on EfficientNet-b0. Fig. 2 shows the two examples with highest distortion on EfficientNet-b0 that still fool SRNet. The added distortion remains imperceptible to the human eye even in these cases.

V. CONCLUSION

This paper explores both sides of adversarial image detection with steganographic glasses.

On the Defense side, we use SRNet [11], state-of-the-art in steganalysis to detect adversarial images. Training it on images

TABLE III
UNDETECTABILITY OF STEGANOGRAPHIC EMBEDDING AGAINST THE NATUREL MODEL (NAT) AND ITS ROBUST VERSION (ROB).

	d	P_{suc} (%)		$\overline{L_2}$		SCRMQ1(%)		SRNet(%)	
		Nat	Rob	Nat	Rob	Nat	Rob	Nat	Rob
[31]	2	88.0	71.8	**84**	**112**	81.2	76.4	93.3	87.5
HILL	2	88.0	71.8	93	117	74.8	66.3	86.1	77.6
HILL	4	**88.8**	**72.6**	105	129	72.4	72.4	85.5	72.3
MiPod	2	87.9	71.8	100	124	74.9	64.3	84.0	76.1
MiPod	4	88.2	72.2	114	137	72	57.0	82.6	67.5
GINA	2	88.0	71.8	168	181	5.4	**3.0**	44.2	33.5
GINA	4	88.2	71.9	232	243	**3.8**	3.1	**20.7**	**14.2**

attacked with the basic FGSM shows impressive performance. Detection also generalizes well even on the finest attacks such as PGD_2 [3] and CW [30].

On the Attack side, our work on steganographic embedding is able to reduce dramatically the detection rates. The steganographic embedding targets specific regions and pixels of an image to quantize the attack. The distortion increases w.r.t. the original attack but remains imperceptible by the human eye (Fig. 2). The main conclusion is that the field of steganography benefits more to the attacker than to the defender.

Our future works will explore the effect of retraining detectors on adversarial images crafted with steganographic embedding towards an even more universal detector.

REFERENCES

[1] S. Ma, Y. Liu, G. Tao, W. Lee, and X. Zhang, "NIC: detecting adversarial samples with neural network invariant checking," in *NDSS 2019, San Diego, California, USA.*, 2019.

[2] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 135–147.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR 2018, Vancouver, BC, Canada.*, 2018.

[4] O. Taran, S. Rezaeifar, T. Holotyak, and S. Voloshynovskiy, "Defending against adversarial attacks by randomized diversification," in *IEEE CVPR*, Long Beach, USA, June 2019.

[5] ——, "Machine learning through cryptographic glasses: combating adversarial attacks by key based diversified aggregation," in *EURASIP Journal on Information Security*, January 2020.

[6] A. Athalye, N. Carlini, and D. A. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *ICML*, 2018, pp. 274–283.

[7] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," *arXiv:1705.07263*, 2017.

[8] E. Quiring, D. Arp, and K. Rieck, "Forgotten siblings: Unifying attacks on machine learning and digital watermarking," in *IEEE European Symp. on Security and Privacy*, 2018.

[9] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv*, 2019.

[10] C. Xie, M. Tan, B. Gong, J. Wang, A. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," *arXiv*, 2019.

[11] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.

[12] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4206–4210.

[13] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *Information Forensics and Security, IEEE Transactions on*, vol. 11, no. 2, pp. 221–234, 2016.

[14] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *Information Forensics and Security, IEEE Trans. on*, vol. 10, no. 9, 2015.

[15] Y. Wang, W. Zhang, W. Li, X. Yu, and N. Yu, "Non-additive cost functions for color image steganography based on inter-channel correlations and differences," *IEEE Trans. on Information Forensics and Security*, 2019.

[16] X. Qiu, H. Li, W. Luo, and J. Huang, "A universal image forensic strategy based on steganalytic model," in *Proc. of ACM IH&MMSec '14*, New York, NY, USA, 2014, pp. 165–170.

[17] S. Farooq, M. H. Yousaf, and F. Hussain, "A generic passive image forgery detection scheme using local binary pattern with rich models," *Computers & Electrical Engineering*, vol. 62, pp. 459 – 472, 2017.

[18] W. Luo, H. Li, Q. Yan, R. Yang, and J. Huang, "Improved audio steganalytic feature and its applications in audio forensics," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 2, Apr. 2018.

[19] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification based machine learning approach with high order statistics features," in *EUSIPCO*, 2016, pp. 1183–1187.

[20] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 3, pp. 868–882, 2012.

[21] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.

[22] Y. Yousfi, J. Butora, J. Fridrich, and Q. Giboulot, "Breaking ALASKA: Color separation for steganalysis in jpeg domain," in *Proc. of ACM IH&MMSec '19*, 2019, pp. 138–149.

[23] W. Fan, K. Wang, and F. Cayre, "General-purpose image forensics using patch likelihood under image statistical models," in *IEEE Int.Workshop on Information Forensics and Security (WIFS)*, 2015, pp. 1–6.

[24] Z. Chen, B. Tondi, X. Li, R. Ni, Y. Zhao, and M. Barni, "A gradient-based pixel-domain attack against svm detection of global image manipulations," in *IEEE WIFS*, 2017, pp. 1–6.

[25] P. Schöttle, A. Schlögl, C. Pasquini, and R. Böhme, "Detecting adversarial examples - a lesson from multimedia security," in *European Signal Processing Conference (EUSIPCO)*, 2018, pp. 947–951.

[26] J. Liu, W. Zhang, Y. Zhang, D. Hou, Y. Liu, H. Zha, and N. Yu, "Detection based defense against adversarial examples from the steganalysis point of view," in *IEEE/CVF CVPR*, 2019, pp. 4820–4829.

[27] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "CNN-based adversarial embedding for image steganography," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2074–2087, 2019.

[28] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR 2015, San Diego, CA, USA,*, 2015.

[29] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger, "Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses," in *Proc. of the IEEE CVPR*, 2019.

[30] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symp. on Security and Privacy*, 2017.

[31] B. Bonnet, T. Furon, and P. Bas, "What if adversarial samples were digital images?" in *Proc. of ACM IH&MMSec '20*, 2020, pp. 55–66.

[32] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[33] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," 2018.

[34] E. Dohmatob, "Generalized no free lunch theorem for adversarial robustness," in *Proc. of Int. Conf. on Machine Learning*, Long Beach, California, USA, 2019.

[35] R. Cogranne, V. Sedighi, J. Fridrich, and T. Pevnỳ, "Is ensemble classifier needed for steganalysis in high-dimensional feature spaces?" in *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*. IEEE, 2015, pp. 1–6.

[36] M. Goljan, J. Fridrich, and R. Cogranne, "Rich model for steganalysis of color images," *2014 IEEE International Workshop on Information Forensics and Security, WIFS 2014*, pp. 185–190, 04 2015.