Evasion Attack STeganography: Turning Vulnerability Of Machine Learning To Adversarial Attacks Into A Real-world Application

Salah GHAMIZI
University of Luxembourg
salah.ghamizi@uni.lu

Maxime Cordy
University of Luxembourg
maxime.cordy@uni.lu

Mike Papadakis
University of Luxembourg
michail.papadakis@uni.lu

Yves Le Traon
University of Luxembourg
yves.letraon@uni.lu

Abstract

Evasion Attacks have been commonly seen as a weakness of Deep Neural Networks. In this paper, we flip the paradigm and envision this vulnerability as a useful application. We propose EAST*, a new steganography and watermarking technique based on multi-label targeted evasion attacks. The key idea of EAST is to encode data as the labels of the image that the evasion attacks produce.*

Our results confirm that our embedding is elusive; it not only passes unnoticed by humans, steganalysis methods, and machine-learning detectors. In addition, our embedding is resilient to soft and aggressive image tampering (87% recovery rate under jpeg compression). EAST outperforms existing deep-learning-based steganography approaches with images that are 70% denser and 73% more robust and supports multiple datasets and architectures.

We provide our algorithm and open-source code at https://github.com/yamizi/Adversarial-Embedding

1. Introduction

Evasion attacks are arguably one of the most actively studied security problems in Machine Learning (ML). Most research focuses on generating an adequate noise to fool an ML-based system [5], understanding the inherent properties that make a model sensitive to such noise [18], and improving the robustness of the model to such perturbations [27].

Hence, research commonly considers that the sensitivity of models to small perturbations is a major weakness that needs to be mitigated. However, some voices also argue that this sensitivity is not only a weakness but also a feature of the models [6, 28]. For instance, Goodfellow et al. [14] were the first to suggest that artificial noise can be designed in a meaningful way to hide and retrieve information.

We argue that an interesting instantiation of this idea is the development of steganography techniques from evasion attacks. Steganography is the process of hiding important information in a trivial medium, e.g., images [4] or audio files [8]. It can be used to transmit a message between a sender and a recipient in a way that a malicious third party cannot detect that the medium hides a message or alter it.

Watermarking is another application case that share the same embedding protocol. It aims to add invisible information into a medium such as only the legitimate recipient can decode it. In steganography, the sender and the recipient are different agents and the purpose is to transmit some secret information. In watermarking the encoder and decoder are generally the same and the purpose is to authenticate the medium with the "noise" (called *watermark*) that has been hidden in the medium. The watermark can be used to authenticate the medium or for Digital Right Management [21]. The images where the perturbation is added are called *cover images* and the images after perturbation are referred to as *stego images*.

Important properties for steganography and watermarking include the elusiveness of the perturbation, the confidentiality of the encoded message, the density of the encoding and the robustness of the stego images to tampering.

Unfortunately, traditional steganography approaches are only effective in low payload density [17] and can easily be detected by steganalysis tools; i.e. detectors tailored to detect stego images. They also rely on static heuristics to decide where and how much perturbation needs to be applied to the image, which makes them easier to detect once the heuristic has been disclosed.

The recent advances in Deep Learning (DL) introduced new approaches for dynamic image steganography, however, these approaches can guarantee only some of the properties – but not all [25]. While most DL-based approaches struggle to reach 0.4 bits per pixels (BPP), Zhang et al.

[48] proposed a high density generative adversarial network (GAN) to build the perturbations. Their density can reach 1 BPP. The success rate of the encoding is, however, dataset specific and drops drastically when the distribution of cover images differs from the optimal training distribution. This GAN-based approach is also vulnerable to image tampering. Zhu et al. [50] focus on robustness to tampering and propose to train an auto-encoder to recover images under noise. The robustness is achieved through adversarial training where the noise is introduced in the training process. While the generated stego images are robust, this technique achieves low density (30 bits in a 128×128 color image). Finally, the above approaches require training expensive models.

In this work, we address these limitations and propose **EAST**, a technique for image stenography and watermarking that taps into the burgeoning field of multi-label evasion attacks. Our approach is plug-and-play and can be used on top of any dataset and deep neural network. It leverages the very well-established properties of evasion attacks to craft steganographic images that are elusive, secure and robust – all this while achieving high density.

We conduct an empirical evaluation to demonstrate the viability and potential of EAST on 2 commonly used computer vision datasets (CIFAR10 and ImageNet). We demonstrate that our technique ensures all the desired properties of a stenographic scheme: *density*, the size of the payload we can correctly decode from a stego image, measured as bits per pixel (BPP); *elusiveness*, the ability to pass unnoticed for a human user and minimally impact the original content; *confidentiality*, its ability to avoid detection by steganalysis tools; and *availability*, the degree to which encoded messages can still be recovered if the stego image has been degraded. Furthermore, we demonstrate that under a similar threat model and protocol, EAST achieves better performances against the state-of-the-art deep-learning-based steganography approaches with images that are up to 70% denser and 73% more robust.

2. Background and related work

2.1. Steganography and Watermarking

Static steganography and watermarking: Traditional techniques use either the spatial domain or the frequency domain to hide information. When operating on the spatial domain, the algorithm changes some pixels on the image to embed data. Common techniques are LSB (Least Significant Bit) [3] and PVD (Pixel-value Differencing) [44]. Advanced techniques like HUGO [31] can hide 7 times longer messages than LSB at an identical level of detectability and WOW [15] uses syndrome-trellis codes to minimize the expected distortion applied to a given payload.

Frequency domain steganography relies instead on frequency distortions [23] to produce the perturbation, such as discrete cosine transform, discrete wavelet transform, and singular value decomposition. The most prominent approaches include J-UNIWARD [16] and F5 [43].

These methods suffer from relying on fixed heuristics to choose the area and amount of perturbation to add to the cover image. It makes the statistics of the collected images easy to detect, extract and reverse engineer.

Machine Learning (ML) then Deep Learning (DL) approaches have been proposed to overcome these limitations and encode the message without any static heuristic.

Deep-learning steganography and watermarking: Tang et al. [42] proposed ADV-EMB, a technique that uses traditional statistical heuristics (UNIWARD) to hide the message then an evasion attack against the detector to update the heuristic to make it less detectable.

While the generated stego images can reach high elusiveness with small messages, its elusiveness drops significantly with the increased density. Contrary to ADV-EMB, we do not use evasion attacks only to fool a detector but rely on the perturbation themselves to carry the data. By doing so, our method achieves both high elusiveness and high density.

With the recent advances in Generative Adversarial Networks (GAN), a new generation of steganography techniques based on GAN has emerged. Volkhonskiy et al. [40] introduced SGAN a three-part GAN where the generator is trained simultaneously with a discriminator (to ensure the realism of cover-images) and a steganalyser (i.e a detector that tries to distinguish clean images and stego images) to ensure elusiveness. SSGAN [36] was later proposed to improve the quality of images generated following the protocol of SGAN. Wang et al. [41] proposed another GAN architecture that generates images that look realistic.

Following a similar architecture, SteganoGAN [48] and HiDDen [50] were proposed. These techniques consist of an Auto-encoder where the critic (a steganalyzer) is used to ensure that the generated images are elusive. SteganoGAN proposes novel deep neural network layers that maximize the density of the payload at the expense of the accuracy of the recovered message. HiDDen proposes to add a differentiable noise layer between the encoder and the decoder and train the whole model at once. This layer ensures that the loss reconstruction takes into account potential perturbations of the images and generates robust stego images. It is however restricted to low-density scenarios and vulnerable to ML steganalysers.

These GAN-based approaches have been designed to optimize a specific property – elusiveness, robustness, or density – but no approach can reach decent levels in all properties. Instead of designing a mechanism from scratch, we suggest building the steganographic approach on top of the rich literature about evasion attacks.Our approach increases

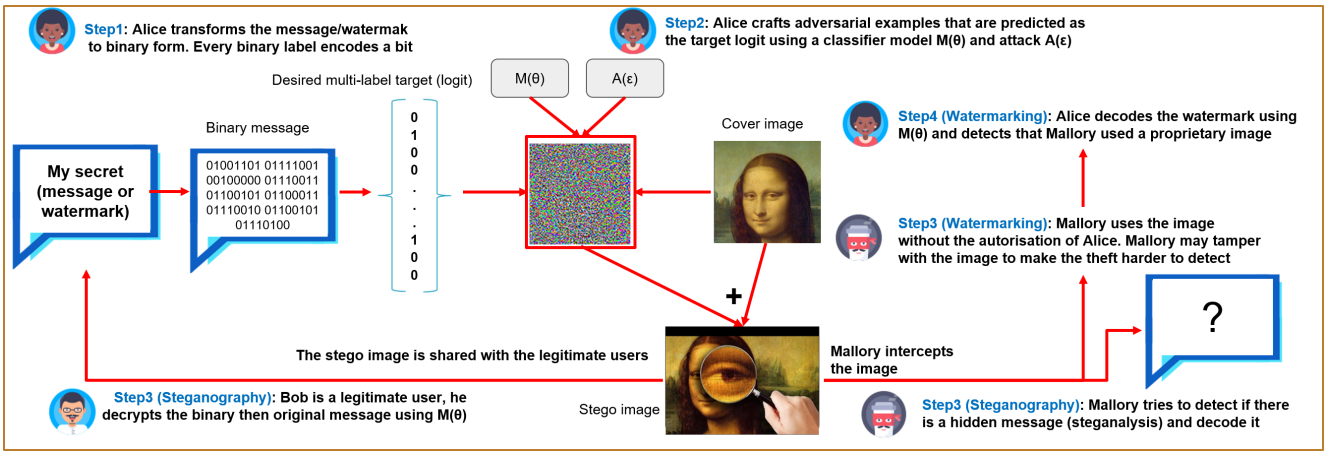

Figure 1: **Sending and decoding a stego image**: When Alice sends a crafted image to Bob, it may be perturbed (benignly or maliciously) during transmission. In the steganography use case, Mallory may try to detect if the image contains any hidden message and Bob can recover the original message. In the Watermarking use case, Mallory may use the image without authorization. Alice can decode the watermark even after tampering and detect the theft.

both the density and security properties of the stego images.

2.2. Evasion attacks

Adversarial examples use intentional minor perturbations to original inputs to alter the prediction of an ML model. Since the first attack algorithms, researchers have played a cat-and-mouse game. They design defense mechanisms such as distillation [29], adversarial training [22], generative adversarial networks [34] to protect the machine learning models. Meanwhile, they also elaborate stronger attack algorithms to evaluate and circumvent these defenses, e.g., PGD [27] and AutoAttack [5].

The literature related to applications of adversarial examples is scarce and mainly focuses on their ability to fool ML-based systems used for, e.g., image recognition [35], malware detection [1] or porn filtering [47]. In this work, we rather consider adversarial examples as a useful means of embedding and hiding secret messages.

Traditional targeted evasion attacks target single-label classifiers: only the most probable class is targeted. Building a steganographic mechanism on top of these attacks is fruitless as it would yield a very low payload density. For instance, attacking the most probable class for an ImageNet classifier with 1,000 classes and 256 image size would result in a density up to $\frac{log(1000)}{log(2)\times 256\times 256} = 1.5\cdot 10^{-4}$BPP (bits per pixels), far from the standards of 0.2 BPP expected from steganography schemes.

Early work on multi-label evasion attacks [37, 45] were able to attack up to 18 labels simultaneously, but already required large distances ϵ and used large image datasets. To achieve competitive payload density, we need to target 20 times more labels. Our approach can attack up to 400 labels simultaneously on small pictures, and reach $\backsim$ 0.4 BPP.

3. Approach

Notations: Assume we have a multi-label classification problem with a set $\mathcal{L}$ of labels. We have $|\mathcal{L}| = m$. Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ a measurable binary multi-label space, with $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{0,1\}^m$, where d is the feature dimension and m is the number of labels. Let function $H : \mathcal{X} \to \mathcal{Y}$ be a multi-label classifier and function $F : \mathbb{R}^d \to \mathbb{R}^m$ be a multi-label predictor that predicts continuous probability scores. These scores indicate the confidence for each label.

Let $F(x) = \{f_1, ...f_m\}$ and $H(x) = \{h_1, ...h_m\}$ $\forall x \in \mathcal{X}$ where $f_j(x)$ is the prediction associated with label $h_j(x)$. Using a classification threshold t, we can induce H from F with, $h_j(x) = \mathbf{I}_{[\![f_j(x)\geq t]\!]}$, where $\mathbf{I}_{[\![.]\!]}$ is an indicator function, that is, $\mathbf{I}$ outputs 1 if the probability score is equal or above the threshold and 0 otherwise.

Let $\mathbf{D} = (x_i, Y_i)_{i=1}^N$ be our dataset with $x_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$. We learn the multi-label predictor $F(x;\theta)$ with parameter θ by solving an optimization problem

$$\min_\theta \frac{1}{N}\sum_{i=1}^{N} l\left(F\left(x_i;\theta\right), Y_i\right), \tag{1}$$

where $l(f(x_i;\theta), Y_i)$ is our multi-label loss function.

3.1. Multi-label evasion attacks

Objectives Given an input x, an evasion attack on a multi-label classifier H under a maximal perturbation ϵ is successful if and only if it produces a perturbation δ such that:

1. $\|\delta\|_p \leq \epsilon$;
2. x and $x+\delta$ share the same ground truth label Y;
3. $H(x+\delta) \neq H(x)$,
   i.e., $\exists i \in \mathcal{L}$ such that $h_i(x+\delta) \neq h_i(x)$.

where i is the index of the targeted task and ϵ the maximum perturbation size using a norm p.

This general definition of evasion attack is **non-targeted** as it is not concerned by the individual state of the labels as long as one label is flipped from its original state.

In our use case, we want to control the value of all labels. Therefore, we aim for **multi-label targeted attacks** on binary classifiers. Let $\mathcal{A} \subset \mathcal{L}$ and $\overline{\mathcal{A}} = \mathcal{L} \setminus \mathcal{A}$. $\mathcal{L}$ is the subset of labels we enforce to be 1 while $\overline{\mathcal{A}}$ is the subset of labels we enforce to be 0. Then a targeted attack is successful iff:

1. $\|\delta\|_p \leq \epsilon$;
2. x and $x + \delta$ share the same ground truth label Y;
3. $\forall (i, j) \in \mathcal{A} \times \overline{\mathcal{A}}$; $h_i(x + \delta) = 1$ and $h_j(x + \delta) = 0$.

Multi-label targeted attacks error optimization A common definition [37] of multi-label targeted evasion attacks minimizes the class prediction error under a fixed budget of perturbation:

$$\begin{aligned} &\underset{\delta}{\text{maximize}} \sum_{i \in \mathcal{A}} \mathbf{I}(h_i(x + \delta) = 1) + \sum_{i \in \overline{\mathcal{A}}} \mathbf{I}(h_i(x + \delta) = 0) \\ &\text{subject to } \|\delta\|_p \leq \epsilon, \end{aligned} \tag{2}$$

where ϵ is a perturbation budget under a norm p and $\mathbf{I}(.)$ an indicator that outputs 1 if the equality $(.)$ inside holds and 0 otherwise. Equation 2 caps the amount of perturbation. It enforces the properties of *Elusiveness* and *Confidentiality* if the ϵ value used in the optimization is low enough.

3.2. EAST: Evasion Attack STeganography

Overview (see Figure 1). Our objective is to design an approach for image watermarking and steganography that relies on the known properties of adversarial examples. Its principle is to encode a message as an adversarial image. More precisely, EAST associates the data to encode with a corresponding set of labels. Each binary label is a bit of the encoded message. Then, our algorithm crafts the adversarial image in such a way that a given multi-label model M_θ classifies the image into the desired labels. To decode the message, the recipient uses the same model to retrieve its set of labels and map these labels back to the original data.

The encoder/decoder: A multi-label model. The model M_θ acts as a secret key for both embedding and decoding. We assume that the model can be transmitted from the sender to the recipient without being intercepted or altered, e.g. through a secure physical device. The model also impacts the success rate of the evasion attack.

EAST can use any multi-label image classification model. Multi-label tasks are generally associated with image segmentation or top-k label prediction. However, common datasets of this task (COCO [24], VOC [11]) only handle few dozen labels. For our approach to achieve high density, we need models with a multiple orders of magnitudes more labels – at least 400 labels in our experiments.

We build these classifiers using CIFAR10 and Imagenet datasets. To obtain a large number of outputs, we slice the last dense layer. In Figure 2, the penultimate layer becomes the logit layer and we include a sigmoid activation then a binary threshold. For instance, a 10-class single label Cifar10 Resnet20 model becomes a 4096 logits multi-label model.

θ denotes the set of parameters that define the model. In DNN classifiers, the parameters include the architecture (number, types, and inner parameters of the layers), the weights of the layers learned during training, the loss, and the regularizers.

The adversarial attack A. The actual generation of the adversarial example can use any existing attack algorithm A. The hyperparameters of the attack include the maximum amount of perturbation allowed on images as well as attack-specific parameters. The choice of the attack algorithm and its parameters impacts the success rate and the elusiveness of the attack. However, this choice does not have to be revealed to the recipient.

The literature of evasion attacks offers a large variety of algorithms. Some are gradient based attacks (FGSM [14], PGD [27], Momentum (MIM) [9], AutoAttack [5]), while others are Evolutionnary (CoEva2 [13], OnePixel [38]).

EAST algorithm. Algorithm 1 formalizes our approach.

First, $buildLogits$ (Line 1) transforms the binary message into the model logits.

Given a set of images, some may produce better stego-images (high success rate with small perturbation) than others. We implement a search restart (Lines 4-5), where we select the cover images from the set best suited to embed the data. The internal procedure $coverPick$ selects the cover images whose original logits are close to the encoded message to bootstrap the search for optimal perturbations.

We build EAST using mechanisms from MIM [9], implemented in the internal procedure $computeAdv$ (line 9). However, the iterative attack gets stuck in a local optimum when the number of labels to flip increases. To overcome this phenomenon of *gradient-lock* (i.e. when the gradient gets almost null because of the number of labels to flip), at each step of the iterative attack, we add a random perturbation inside a sphere of starting radius δ around the gradient-based adversary. To ensure convergence, we linearly reduce this radius δ at each iteration step.

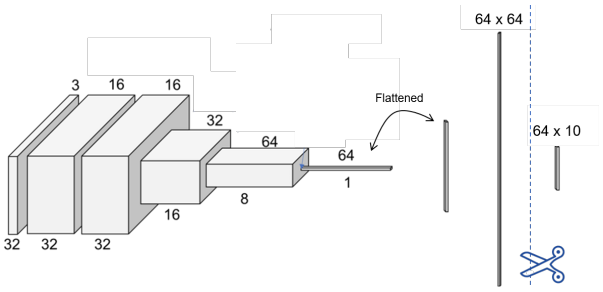

Figure 2: **A typical architecture of an image classification CNN**: A classification neural network ends a series of convolution and pooling layers with a set of dense layers. In EAST, The last dense layer (with the classification logits, 64×10) is removed after the model has been trained.

Error Correcting Codes: While traditional techniques ensure 100% success rate in the encoding and decoding of messages, deep learning techniques may not always achieve perfect performances. This is the case for both GAN-based techniques [48] and our technique based on evasion attacks. Given a maximum budget of perturbation ϵ, the attack can fail to successfully flip all required labels, and the recovered message may contain errors.

We, therefore, rely on error correction codes, a signal processing technique to correct errors. In short, this technique builds a reduced dictionary containing only words with maximum separability. Then, it rewrites the message only with the words of this reduced dictionary. With a lookup into the dictionary, it can guess which words have been corrupted (words of the message not in the dictionary) and infer the original words. Error correction algorithms differ in the way they build the dictionary, and how they evaluate the corrupted words and recover the original ones.

In our approach, we use Reed–Solomon error correction algorithm [33] (RS-Codes), which is one of the most common error correction technique. RS-Codes can transform a message M_l of length l into a message M_m of length m such that it can recover the original message M_l if the new message M_m has at most $\frac{m-l}{2}$ errors.

Given the success rate of the evasion attack s, RS-Codes on a message of length m can recover the message M_l if $(1-s).m \leq \frac{m-l}{2}$. As a result the message correctly recovered is as long as $l \leq (2s-1).m$. In practice, this means that if our attack has a success rate of 90% when encoding a message with 1 BPP density, the actual message that could be recovered without error is only of a density of 0.8 BPP.

4. Experiment

4.1. Metrics

BPP: Bits Per Pixels reflects the density of the embedding. It is computed as the ratio of total size of the encoded message (# bits) over the size of the image (width . height).

Algorithm 1: EAST algorithm

input : A classifier M_θ; A dataset of cover images I and encoded message D_{enc}; step size ϵ_{step}; maximum perturbation size ϵ; initial random sphere size δ; total iterations L; Number of cover restarts R;

output: $bestAdv$: The stegano-images that encode D_{enc}

$logits \leftarrow$ `buildLogits(`D_{enc},M_θ`)`;
$bestRate \leftarrow 0$;
$bestAdv \leftarrow Null$;
for $j \leftarrow 1$ **to** R **do**
  $I_{start} \leftarrow$ `coverPick(`I,$logits$`)`;
  $advX \leftarrow I_{start}$;
  $momentum \leftarrow 0$;
  for $i \leftarrow 1$**to**L **do**
    $advX, momentum \leftarrow$ `computeAdv(`$advX$, $logits$, I_{start}, ϵ_{step}, ϵ, $momentum$`)`
    $advX_{noised} \leftarrow$ `randomSphere(`$advX$, i, δ`)`
    $success \leftarrow$ `computeSuccess(`M_θ, $advX$, $logits$, m`)`;
    $success_{noised} \leftarrow$ `computeSuccess(`M_θ, $advX_{noised}$, $logits$`)`;
    if $success \leq success_{noised}$ **then**
      $success \leftarrow success_{noised}$;
      $advX \leftarrow advX_{noised}$;
    end
    if $bestRate \leq success$ **then**
      $bestRate \leftarrow success$;
      $bestAdv \leftarrow advX$;
    end
  end
end

BSR: Bits Success Rate is a natural performance metric to evaluate the quality of the embedding. It is the percentage of bits that are correctly decoded from a stego-image.

RS-BPP: Reed-Salomon Bits Per Pixels is the metric derived from the Error Correcting Codes and the BSR, and we have: RS-BPP $= (2.\text{BSR} - 1).\text{BPP}$

SSIM: Structural Similarlity Index Metric [49] roughly measures how close two images are. It is known to be a better metric than others like signal-to-noise ratio (PSNR) and mean squared error (MSE) [12].

SSIM can be expressed between two images x and y as

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3)$$

where μ_x and μ_y the mean of x and y respectively; σ_x and σ_y the variance of x and y respectively and σ_{xy} their associated covariance; C_1 and C_2 stabilize the division;

Requirement	Watermark	Steganography
Effectiveness (RQ1)	✓	✓
Elusiveness (RQ2)	✓	✓
Confidentiality (RQ3)		✓
Availability (RQ4)	✓	

Table 1: Expected properties for both use cases of watermarking and steganography.

4.2. Research Questions

Our evaluation assesses whether EAST can serve as a secure steganography and watermarking technique.

Table 1 highlights the security requirements for the two applications. Effectiveness refers to the ability to perform the embedding and reach a high density. Both applications require that the embedded message be *elusive* – i.e. the message should not alter the media and the perturbation should not be visible. The difference between the two applications lies in that steganography expects the message to remain confidential and any third party should not be able to detect the message and apply steganalysis techniques. On the other hand, watermarking requires resilience to data destruction, i.e., the *availability* of the encoded information.

Our first research question is to check if EAST is a viable approach for steganography:

RQ1 *What is the success rate and density that EAST achieves?*

We assess the success rate (BSR) and density (RS-BPP) of EAST under different perturbation budgets ϵ. Density is an important property to evaluate, which is especially relevant for steganography. We, therefore, run our experiments under two payload capacities: 410 bits and 1024 bits. 410 bits represent a density of 0.4 BPP – a typical expectation for practical stenography approaches [17] and 1024 bits represent a density of 1 BPP – the highest density achieved by state of the art techniques (SotA) [48]. We compare, for each message size, the performance of EAST with the SotA.

Our next research question focuses on the ability of EAST to avoid being noticed. Thus, we ask:

RQ2 *Does EAST produce elusive perturbations?*

We evaluate elusiveness by measuring the similarity – with the SSIM metric – between the cover image and the image that EAST produces. We do this measurement for multiple ϵ values and embedding densities. We also evaluate the tradeoff between success rate and similarity.

Elusiveness is not sufficient, especially in the use case of steganography where detectors exist. We, therefore, check the confidentiality of EAST, that is, whether steganalysis can detect EAST's adversarial examples. We ask:

RQ3 *Can EAST be detected by steganalysis?*

We assess the detectability of the cover images against 2 detectors. The first is Sample Pair Analysis (SPA) [10] and it is one of the most common statistical steganalysis techniques. Statistical steganalysis recognizes the noise profile of different steganography techniques.

The second detector is the ML steganalysis approach of Ye et al [46]. This is the state-of-the-art approach and it supports different threat models. In our experiments, we evaluate three threat models: blackbox (the interceptor has access to unrelated sets of perturbed images and clean images), graybox (the interceptor has access to pairs of clean and perturbed images but does not know the hyperparameters of the generation process), and whitebox (the interceptor has full access to the generation process).

After studying the confidentiality of the embedded information, we turn our attention towards its availability when reaching the recipient. Availability is threatened by malicious image degradation. Hence, we want to ensure that decoding tampered images still yields the original message. We consider *spatial domain* degradation resulting from basic image transformations (rotation, upscaling, and cropping) as well as *frequency domain* degradation like JPEG compression and color depth reduction. This property is critical for the Watermarking use case. Thus, we ask:

RQ4 *Does EAST resist to image tampering?*

First, we focus on *Spatial Domain degradation* and three local image alterations: *rotation* (we rotate the images by 15°); *upscaling* (bilinear interpolation to resize the images to 64x64 pixels); *cropping* (removing 12.5% of the images, keeping only the central part). These transformations are common when copyrighted images are shared illegally [32].

Second, we study the impact of *Frequency Domain degradation*: JPEG compression and Color Depth Reduction (CDR). JPEG compression relies on various steps (color transformation, DCT, quantization) that cause information loss. CDR reduces the number of bits used to encode different colors. We apply JPEG compression with 90% and 50% quality rates (resulting in loss of information of 10% and 50%, respectively) and CDR (to 8 bits, i.e. pictures with only 1/12 of the original information).

We measure for each transformation the recovery rate of the transformed images; i.e. the percentage of images where the corrupted and original images have the same label.

4.3. Experimental setting

The experiments were performed on a Tesla V100-SXM2-32GB GPU on an Nvidia DGX. The EAST algorithm is implemented on top of the torchattacks [19] library. The RS-Codes rely on the python implementation of Reed-Solomon Codes [39]. The comparison with the SotA

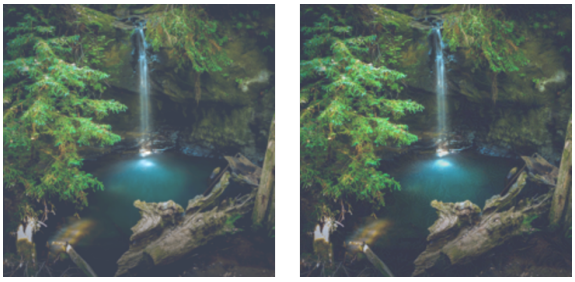

Figure 3: Left: Cover image, Right: Stego image; From ImageNet; SSIM 99.8%

is done using the authors' original library. To implement EAST, we use pre-trained Pytorch Zoo [30] Resnet20 on CIFAR10 [20] and Resnet18 for ImageNet [7].

We evaluate EAST under a maximum budget of 500 iterations and perturbation size of $\epsilon \in \{8/255, 16/255, 0.1, 0.3, 0.5, 1\}$ to cover all the possible range under ℓ_∞ norm. We target an embedding density of 0.4BPP (410 bits) and 1BPP (1024 bits). We use a momentum of 1 and we linearly decrease the size of the random dual sphere starting with a radius of $10^{-2} \cdot \epsilon$.

Baselines and State of the Art Liu et al. [26] provide an extensive evaluation of DNN and GAN-based steganography. These techniques suffer from a low embedding capacity and will not be evaluated in this study.

Instead, we evaluate our approach against the highest density technique available in the literature. This technique is *STEGANO-GAN* [48], a novel steganography scheme that achieves a density of up to 4 BPP on some specific datasets.

5. Results and analysis

5.1. RQ1: Effectiveness

Table 2 shows that EAST can reliably encode and decode messages, with a bit success rate up to 86.2% and a message density of 0.373 BPP.

Besides, EAST does not suffer from a large drop in success rate (BSR) when embedding more bits: Moving from 0.4 BPP (410 bits, i.e. 410 labels) to 1 BPP (1024 labels) causes only a slight drop under a small perturbation budget $\epsilon = 8/255$: The success rate drops from 0.586 under 0.4 BPP to 0.528 under 1BPP. Similarly, it drops from 0.862 to 0.692 under a high perturbation budget.

This last result hints that building neural networks with a larger logits layer can lead to ever-increased data density with limited impact on the success rate.

5.2. RQ2: Elusiveness

We also compare in 2 the performance of EAST with the SotA "SteganoGAN" both in terms of density and image similarity. Under similar RS-BPP (0.219 for SteganoGAN and 0.260 for EAST), our approach achieves an image similarity (SSIM) up to 0.931 while SteganoGAN is capped at 0.466. Our encoded images are imperceptible, even for a human operator. Our approach also achieves a 0.373 RS-BPP given a sufficient perturbation budget ($\epsilon = 1$).

Comparing the hyper-parameters of EAST, there is a significant tradeoff between visual similarity and density: The average similarity of the encoded images drops from 0.771 under a perturbation size of $\epsilon = 8/255$ to 0.308 under a perturbation size of $\epsilon = 1$. However, using the cover pick strategy introduced in Algorithm 1, we can find cover images that achieve 0.689 SSIM under a perturbation size of $\epsilon = 1$. This confirms that EAST can allow both high density and similarity if the sender has access to a pool of cover images and the algorithm can pick the best ones.

5.3. RQ2: Confidentiality

Table 3 confirms that our approach and the SotA are both resilient to statistical analysis (SPA). While our approach resists well to blackbox and graybox ML steganalysis (with a respective detection rate of 0.571 and 0.682), SteganoGAN is always detected under these threat models.

EAST resists whitebox steganalysis under small perturbation budgets. However, adding more noise makes EAST vulnerable under this threat model. The resilience of EAST to Whitebox detectors can be explained by the inherent randomness introduced by EAST thanks to the random perturbations at each step of our iterative attack (random exploration) and the stochastic nature of gradient descent.

Our results show that the sweet spot for steganalysis is when the images are encoded with a medium perturbation budget. The steganalyser of Ye et al. [46] may have a capacity tailored to this amount of perturbation, and its performance degrades when the perturbation size increases.

5.4. RQ3: Availability

We evaluate spatial-domain tampering in Table 4 and frequency-domain tampering in Table 5. While images embedded using SteganoGAN fail to recover the original message (up to 51% bits are lost when the images are rotated), EAST provides high robustness against spatial tampering. Up to 88.9% of the bits in the rotated images are identical to the non-tampered images. EAST shows similar robustness against Scaling and Cropping. Increasing the amount of perturbation has little impact on EAST's robustness to spatial tampering. While up to 88.9% of the bits are recovered from rotated images with small perturbation ($\epsilon = 8/255$), it only drops to 84.0% under $\epsilon = 1$.

EAST is also more robust than SteganoGAN against image JPEG compression and bit depth reduction. While only up to 54.6% of the bits are preserved with SteganoGAN, EAST can preserve up to 89.1% of the bits when the depth is reduced and up to 86.7% when compressed at 50%.

Algorithm		410 bits / 0.4 BPP				1024 bits / 1 BPP				RS-BPP$_{max}$
	ϵ	$SSIM_{max}$	$SSIM_{mean}$	BSR_{max}	BSR_{mean}	$SSIM_{max}$	$SSIM_{mean}$	BSR_{max}	BSR_{mean}	
EAST	8/255	**0.995**	0.771	0.586	0.507	**0.994**	**0.758**	0.528	0.490	0.068
	16/255	0.989	**0.756**	0.648	0.549	0.989	0.756	0.556	0.508	0.117
	0.1	0.984	0.743	0.675	0.583	0.984	0.744	0.573	0.526	0.143
	0.3	0.930	0.645	0.786	0.686	0.931	0.646	0.634	0.578	0.260
	0.5	0.865	0.534	0.828	0.731	0.864	0.535	0.662	0.601	0.314
	1	0.689	0.308	**0.862**	**0.782**	0.689	0.323	**0.692**	**0.639**	**0.373**
SteganoGAN	-	0.194	0.118	0.620	0.604	0.466	0	0.623	0.603	0.219

Table 2: Effectiveness and elusiveness, for 0.4 BPP and 1 BPP encoding. Best values in bold.

Algorithm		SPA	Ye et al. Steganalysis		
	ϵ		Blackbox	Graybox	Whitebox
EAST	8/255	0.5	0.571	0.682	0.75
	16/255	0.5	0.707	0.921	0.964
	0.1	0.5	0.842	0.967	0.979
	0.3	0.5	0.981	0.905	0.968
	0.5	0.5	1	0.977	0.994
	1	0.5	0.633	0.833	0.992
SteganoGAN	-	0.5	1	1	1

Table 3: AUC of Steganalysis detector at 1BPP

Algorithm		Rotation		Scaling		Cropping	
	ϵ	mean	max	mean	max	mean	max
EAST	8/255	0.767	0.887	0.767	0.880	0.720	0.833
	16/255	0.740	0.866	0.733	0.887	0.698	0.819
	0.1	0.714	0.863	0.708	0.871	0.681	0.798
	0.3	0.658	0.831	0.650	0.861	0.639	0.716
	0.5	0.633	0.840	0.630	0.818	0.593	0.707
SteganoGAN	-	0.482	0.513	0.582	0.634	0.504	0.524

Table 4: Robustness against spacial-domain degradation

Algorithm		Depth		JPEG 90		JPEG 50	
	ϵ	mean	max	mean	max	mean	max
EAST	8/255	0.774	0.879	0.768	0.864	0.771	0.867
	16/255	0.749	0.891	0.738	0.842	0.736	0.854
	0.1	0.717	0.869	0.708	0.845	0.710	0.825
	0.3	0.656	0.857	0.650	0.838	0.652	0.835
	0.5	0.639	0.817	0.633	0.822	0.631	0.785
SteganoGAN	-	0.521	0.546	0.512	0.520	0.490	0.510

Table 5: Robustness against frequency-domain degradation

6. Limitations and opportunities

Unlike existing steganography techniques, EAST is not impacted by the size of the input images but by the size of the logits layer of the neural network.

In Table 6 we show that we achieve a better success rate with a model tailored for ImageNet. For example, Figure 3 shows a combination of cover and stego images that achieves high elusiveness with a reliable success rate on ImageNet. The model has a 6 times larger logits layer but requires images 64 times bigger than Cifar10 (256×256 vs 32×32). The larger size of images decreases the RS-BPP of EAST. To offset this larger size, we can increase the size of the logit layer by the same factor. Hence, for a fair comparison, we evaluated our research questions under the same testbed and experimental setting for EAST and the SotA.

Another benefit of EAST is that it immediately benefits from the advances in evasion attack research. Techniques that propose stronger attacks with smaller perturbations can substantially improve EAST. Additionally, gradient attacks can be designed in an adaptive way: their loss functions are tuned to take into account specific objectives (resilience to steganalysis, robustness to compression/feature squeezing as proposed by [2]). This strong link with evasion attack research gives EAST a substantial competitive advantage against all the other techniques. These techniques require their own line of research to achieve similar improvements.

	Elusiveness			Steganalysis		
ϵ	SSIM	BSR	RS-BPP	Black	Gray	White
8/255	0.998	0.742	0.009	0.625	0.958	0.970
1	0.790	0.974	0.019	0.974	0.989	0.989
Domain	Spatial Tampering			Frequency Tampering		
ϵ	Rotate	Scale	Crop	Depth	JPEG 90	JPEG 50
8/255	0.721	0.799	0.686	0.781	0.800	0.799
1	0.527	0.630	0.535	0.569	0.585	0.565

Table 6: Evalution of EAST on ImageNet model at 1024bits

7. Conclusion

We proposed EAST, a new technique to hide secret messages in images using evasion attack algorithms. We proposed an algorithm for large-scale multi-label targeted evasion attacks and demonstrated that EAST is an effective, elusive, secure, and flexible technique for steganography and watermarking . We have also shown that our multi-label evasion attack, combined with appropriate DNN, enables large payloads embedding while preserving the security criterion and significantly outperform SotA on all criteria.

An inherent benefit of our approach is that it leverages targeted evasion attack algorithms and research. Therefore, our technique can take advantage of any future development coming out from this highly active research area.

Evasion attacks are not restricted to vision tasks, and future work should expand our study to larger models and tasks. In particular, other media where adversarial examples have shown mature results such as audio, video, and text can be transformed using EAST into means of hiding data, with tangible industrial applications for digital right management and privacy.

References

[1] Hyrum S. Anderson. Evading machine learning malware detection. 2017.

[2] Nicholas Carlini and David Wagner. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. 2017.

[3] Chi-Kwong Chan and L.M. Cheng. Hiding data in images by simple lsb substitution. *Pattern Recognition*, 37:469–474, 03 2004.

[4] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. *Signal Processing*, 90(3):727 – 752, 2010.

[5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. 2020.

[6] Pieter Delobelle, Paul Temple, Gilles Perrouin, B. Fr'enay, P. Heymans, and B. Berendt. Ethical adversaries: Towards mitigating unfairness with adversarial machine learning. *ArXiv*, abs/2005.06852, 2020.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] F. Djebbar, B. Ayad, H. Hamam, and K. Abed-Meraim. A view on latest audio steganography techniques. In *2011 International Conference on Innovations in Information Technology*, pages 409–414, April 2011.

[9] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. 2018.

[10] Sorina Dumitrescu, Xiaolin Wu, and Zhe Wang. Detection of lsb steganography via sample pair analysis. In *International workshop on information hiding*, pages 355–372. Springer, 2002.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[12] Jeremy R. Flynn, Steve Ward, Julian Abich, and David Poole. Image quality assessment using the ssim and the just noticeable difference paradigm. In Don Harris, editor, *Engineering Psychology and Cognitive Ergonomics. Understanding Human Cognition*, pages 23–30, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[13] Salah Ghamizi, Maxime Cordy, Martin Gubri, Mike Papadakis, Andrey Boystov, Yves Le Traon, and Anne Goujon. Search-based adversarial testing and improvement of constrained credit scoring systems. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, page 1089–1100, New York, NY, USA, 2020. Association for Computing Machinery.

[14] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. pages 1–11, 2014.

[15] Vojtech Holub and Jessica J. Fridrich. Designing steganographic distortion using directional filters. *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 234–239, 2012.

[16] Vojtech Holub, Jessica J. Fridrich, and Tomás Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014:1–13, 2014.

[17] Mehdi Hussain, Ainuddin Wahid Abdul Wahab, Yamani Idna Bin Idris, Anthony T.S. Ho, and Ki Hyun Jung. Image steganography in spatial domain: A survey. *Signal Processing: Image Communication*, 65(December 2017):46–66, 2018.

[18] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

[19] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks, 2021.

[20] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[21] William Ku and Chi-Hung Chi. Survey on the technological aspects of digital rights management. In *International Conference on Information Security*, pages 391–403. Springer, 2004.

[22] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *ArXiv*, abs/1611.01236, 2016.

[23] Bin Li, Junhui He, Jiwu Huang, and Y.Q. Shi. A survey on image steganography and steganalysis. *Journal of Information Hiding and Multimedia Signal Processing*, 2, 05 2011.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. 2014.

[25] Jia Liu, Yan Ke, Zhuo Zhang, Yu Lei, Jun Li, Minqing Zhang, and Xiaoyuan Yang. Recent advances of image steganography with generative adversarial networks. *IEEE Access*, PP:1–1, 03 2020.

[26] Jia Liu, Yan Ke, Zhuo Zhang, Yu Lei, Jun Li, Minqing Zhang, and Xiaoyuan Yang. Recent advances of image steganography with generative adversarial networks. *IEEE Access*, PP:1–1, 03 2020.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. pages 1–27, 2017.

[28] David J. Miller, Zhen Xiang, and George Kesidis. Adversarial learning in statistical classification: A comprehensive review of defenses against attacks. *ArXiv*, abs/1904.06292, 2019.

[29] Nicolas Papernot, Patrick Mcdaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. *Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016*, pages 372–387, 2016.

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Al-

ban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[31] Tomáš Pevný, Tomáš Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In Rainer Böhme, Philip W. L. Fong, and Reihaneh Safavi-Naini, editors, *Information Hiding*, pages 161–177, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[32] Christine I Podilchuk and Edward J Delp. Digital watermarking: algorithms and applications. *IEEE signal processing Magazine*, 18(4):33–46, 2001.

[33] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[34] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *ArXiv*, abs/1805.06605, 2018.

[35] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. pages 1528–1540, 10 2016.

[36] Haichao Shi, Jing Dong, Wei Wang, Yinlong Qian, and Xiaoyu Zhang. Ssgan: Secure steganography based on generative adversarial networks. 2018.

[37] Qingquan Song, Haifeng Jin, Xiao Huang, and Xia Hu. Multi-label adversarial perturbations. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1242–1247, 2018.

[38] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, Oct 2019.

[39] Filiba Tomer. Reed solomon. `https://github.com/tomerfiliba/reedsolomon`, 2020.

[40] Denis Volkhonskiy, Ivan Nazarov, and Evgeny Burnaev. Steganographic generative adversarial networks. 2019.

[41] YANG Xiaoyuan WANG Yaojie, NIU Ke. Information hiding scheme based on generative adversarial network. *Journal of Computer Applications*, 38(10):2923, 2018.

[42] Tang Weixuan, Bin Li, Shunquan Tan, Mauro Barni, and Jiwu Huang. Cnn based adversarial embedding with minimum alteration for image steganography. 03 2018.

[43] Andreas Westfeld. F5—a steganographic algorithm. In Ira S. Moskowitz, editor, *Information Hiding*, pages 289–302, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[44] Da-Chun Wu and Wen-Hsiang Tsai. A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters*, 24(9):1613 – 1626, 2003.

[45] Z. Yang, Yufei Han, and X. Zhang. Characterizing the evasion attackability of multi-label classifiers. In *AAAI*, 2021.

[46] Jian Ye, Jiangqun Ni, and Yang Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, 2017.

[47] Kan Yuan, Di Tang, Xiaojing Liao, and Xiaofeng Wang. Stealthy Porn : Understanding Real-World Adversarial Images for Illicit Online Promotion. *2019 IEEE Symposium on Security and Privacy*, pages 547–561, 2019.

[48] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. Steganogan: High capacity image steganography with gans. 2019.

[49] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[50] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. 2018.