Are Deepfakes a Game Changer in Digital Images Steganography Leveraging the Cover-Source-Mismatch?

Arthur Méreur, Antoine Mallet, Rémi Cogranne

Are Deepfakes a Game Changer in Digital Images Steganography Leveraging the Cover-Source Mismatch?

Arthur Méreur*
arthur.mereur@utt.fr
Troyes University of Technology
Troyes, France

Antoine Mallet
antoine.mallet@utt.fr
Troyes University of Technology
Troyes, France

Rémi Cogranne†‡
remi.cogranne@utt.fr
Troyes University of Technology
Troyes, France

ABSTRACT

This work explores the potential of synthetic media generated by the means of Artificial Intelligence (AI), sometimes referred to as *Deepfakes*, as a source of cover-objects for steganography. *Deepfakes* offer a vast and diverse pool of media, potentially improving steganographic security by leveraging cover-source mismatch, a challenge in steganalysis where training and testing data come from different sources. The present paper proposes an initial study on Deepfakes' effectiveness in the field of steganography. More precisely, we propose an initial investigation to assess the impact of *Deepfakes* on image steganalysis performance in an operational environment. Using a wide range of image generation models and state-of-the-art methods in steganography and steganalysis, we show that *Deepfakes* can significantly exploit the cover-source mismatch problem but that mitigation solutions also exist. The empirical findings can inform future research on steganographic techniques that exploit cover-source mismatch for enhanced security.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → *Machine learning*; **Computer vision**; *Simulation evaluation*;

KEYWORDS

Steganography, Steganalysis, DeepFakes, Robust detection, Cover-Source Mismatch, Statistical detection

ACM Reference Format:
Arthur Méreur, Antoine Mallet, and Rémi Cogranne. 2024. Are Deepfakes a Game Changer in Digital Images Steganography Leveraging the Cover-Source Mismatch?. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30-August 2, 2024, Vienna, Austria.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3664476.3670893

*All authors contributed equally.
†Corresponding Author.
‡This work has been funded by the EU's Horizon 2020 program under grant agreement No. 101021687 (UNCOVER project), and the French ANR PACeS project No. ANR-21-CE39- 0002.

ARES 2024, July 30-August 2, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07…$15.00
https://doi.org/10.1145/3664476.3670893

1 INTRODUCTION

This digital era has given birth to modern steganography: that is, it enabled methods for concealing a message within a seemingly innocuous digital carrier, the so-called cover. Among all possible covers, digital media are extremely suitable for covering sensitive data due to their redundancy and relatively high content complexity, making it possible to "hide in plain sight", without raising suspicions. In this field, digital images have long been one of the most popular mediums. Indeed, digital images are massively shared over the Internet, the standard JPEG is extremely dominant and, despite the compression, images are simple enough to be easily manipulated for hiding data. In addition, this type of media offers a rather large space for payload sizes of practical interest [18]. For all these reasons, digital images perfectly fit the needs of steganography. In the landscape of communication secrecy and digital forensics, steganography, as a method for communicating with secrecy and imperceptibility, occupies a unique position. While its use can be backed with legitimate reasons, its abuse by terrorist organizations and criminal rings makes it a potential serious threat vector as it is easy to use as leverage to facilitate illicit activities.

The proliferation of steganography software readily available from the Internet and application repositories[1], along with the increasing use of digital images, underscores the critical need for steganalysis which is the counterpart of steganography: it aims at distinguishing genuine digital photography from those containing an embedded message [5].

Steganography and steganalysis thus constitute a game of cat-and-mouse in which the steganographers wish to hide sensitive data into the cover object while the steganalyst aims at detecting the presence of hidden data. Having opposing objectives, they both seek to defeat their adversary or, more precisely, make their task as difficult as possible.

1.1 Advances in Digital Media Steganography and Steganalysis

On the one hand, steganography has been considerably improved thanks to the use of linear error-correcting codes from information theory. First, the use of Huffman codes [41] allowed reducing the number of changes, for the same payload (usually measure in Bits Per Pixels or bpp), hence making detection harder. Then the use of convolutional codes, especially the Syndrome-Trellis Codes [17], brought a dramatic improvement. Not only does it significantly improve the coding efficiency, closing up the gap with Shannon's theoretical bounds, but it mainly allows assigning a cost associated

[1]Dozens of software are available "off-the-shelf" on Android Market, Apple Store as well as various Linux distributions repos simply searching for keyword "steganography" or "image steganography".

with the modification of every element (either pixels or DCT coefficients from JPEG images). This work opened the doors for adaptive steganography that, using a so-called cost function, embeds the sensitive data precisely where detection is expected to be harder. Almost all recent steganography approaches for digital images are based on this principle.

On the other hand, recent breakthroughs in the field of artificial intelligence have developed steganalysis as never before. Steganalysis techniques were initially developed to catch specific traces of steganography [5], but machine learning quickly helped to design universal steganalysis methods that are effective against a wide range of steganographic techniques. Larger and larger handcrafted features sets [14, 19, 34] associated with dedicated classifiers operating on very high-dimensional features space [8, 13, 28], have for instance been very successful during the BOSS challenge and the and the subsequent years [3]. Recently, the introduction of deep learning techniques, with their unparalleled capacity to optimize jointly the feature extraction, pattern recognition and classification tasks, has further improved steganalysis significantly [7, 42, 43], as shown within the ALASKA contest for instance [9, 10].
Furthermore, a new dimension in the interplay between steganography and steganalysis has been introduced with the advent of generative adversarial networks (GANs).

Unfortunately, steganalysis detectors suffer greatly from the so-called Cover-Source Mismatch (CSM) [16, 29, 38]. This phenomenon occurs potentially for all machine-learning-based detectors as it originates from the discrepancy between the statistical properties of images used for training the detector and the images used during testing. In practice, the impact of the CSM is that the performance of a detector can be greatly affected when evaluated over images that are not coming from the same source as the training set. Steganalysis is notably sensitive to the CSM because it seeks to detect a very weak signal within a complex object whose properties can greatly change depending on the many factors involved in its acquisition such as the light source, sensor, ISO sensitivity, in-camera processing, and compression to cite a few. For an in-depth review of both sources and the impact of the CSM in digital image steganalysis, the reader is referred to [32].

1.2 Contribution and Organization of the Present Paper

Since the most reliable method of steganalysis is signature-based, a large part of the security of steganography relies on "implementation details". Among those, one of the golden rules of steganographic security is to not use a cover that can be available to the steganalyst. Generally, it is thus recommended to use digital photography made for the sole purpose of steganography and to delete the cover after it has been used.

However, even in this setting, within the framework of Cover-Source mismatch, the steganographer can pick the cover image among all the sources it has at its disposal. Recently *Deepfakes*, which can be roughly defined as hyperrealistic synthetic media generated by AI methods, have raised widespread attention due to their immediate availability while they reach unprecedented content realism.
Within this context, one intriguing development in this domain, that has never been studied, is the potential utilization of *Deepfakes* for steganography. Indeed, *Deepfakes* can be used as cover images for steganography, potentially offering a large set of sources to any steganographer who can thus fully leverage the Cover-Source mismatch.

Therefore, *Deepfakes*, with their ability, bring a new twist to the cat-and-mouse game of steganography and steganalysis. This paper is the first to study the potential use of *Deepfakes* in steganography and their ability to increase the security of hidden data by leveraging the Cover-Source mismatch. We present the results of a rather empirical study in which we compare the performance of the state-of-the-art steganalysis method over digital photography and *Deepfakes*. In particular, we assess the cover-source mismatch between the various AI methods for generating *Deepfakes* and with natural photography. Recently, it has become clear that CSM is in large part due to the discrepancy of various cover image sources in terms of noise characteristics; we explore how much the observed results are aligned with those recent findings.

The present paper is organized as follows:

First, Section 2 provides a brief overview of the CSM, explaining its origins and the existing approaches to assess its practical impact and mitigate it. Second, Section 3 presents the experimental methodology proposed to study the impact of *Deepfakes* in image steganography. Section 4 presents and discusses the numerical results. Finally, Section 5 concludes the present paper and sketches the plan for possible future works.

2 COVER-SOURCE MISMATCH AND ITS IMPLICATION

In academic studies, steganalysis is very often carried out with knowledge of the properties of the inspected objects, that is the "*source*" from which cover objects are generated. This especially includes the embedding method, the payload, and access to large datasets of objects from the very same "*source*" of covers. However, in real-world scenarios, the steganographer and steganalyst have limited access to each other's information. Under this scenario, it can be argued that academic steganalysis mostly serves to assess steganography in a "worst-case scenario", invoking Kerchoff's principles. On the opposite, the application of academic studies in real-life, operational, contexts gives birth to the so-called cover-source mismatch problem. This problem occurs when the steganalyst trains a detector on a different cover-source than the one used by the steganographer. This mismatch can lead to a catastrophic deterioration in the performance of steganalysis models. In the broad field of statistical learning, this phenomenon referred to as the *distribution shift*, is a common challenge in machine learning applications. However, steganography and digital forensics face unique challenges due to the weak signal of interest and the strong impact of cover-source mismatch, often resulting in ineffective steganalysis.

The CSM problem in steganalysis has been identified for almost 20 years, with the understanding that different datasets yield different performance results [25, 27]. The significance of CSM was recognized during the BOSS contest in 2010, coinciding with the rise of machine learning in steganalysis. However, the problem was rarely directly studied until 2018, when its causes were thoroughly

examined [15, 16]. Even deep learning models were found to be susceptible to CSM. Several strategies have been suggested to mitigate its impact on steganalysis. Our previous paper [30] for instance shows that the software used for data hiding can also give birth to a tremendous mismatch. Despite its severity, CSM remains largely unexplored [26] and is only briefly discussed in existing survey papers. The reader can read [32] for a detailed recent systematic review on CSM, its causes, and its impact on steganalysis. In this work, we define a **cover-source** as all the steps and parameters used in the generation and processing pipeline, based on the findings of this paper. The objects from a cover-source thus share common statistical characteristics.

The ensuing steganalysis problem of cover-source mismatch (CSM problem) is the degradation of a stego-detection performance when training and testing sets come from different cover-sources.

2.1 Assessment of the CSM problem

Before explaining how the cover-source mismatch is assessed, let us formally define the concepts one needs to know, and introduce the notation used to this end.

We use the statistical hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ to represent images drawn from cover-objects and from stego-images respectively. The images are denoted $\mathbf{X}$; they are made of N pixels, hence lie in the space of all possible media $\mathbf{X} \in \mathcal{X} \subset \mathbb{N}^N$.

A scoring function λ is a mapping $\mathcal{X} \to \mathbb{R}$ which assigns to an input image $\mathbf{X}$ a score revealing how much it is supposedly a cover or a stego-object. The ensuing detector δ is a mapping $\delta : \mathcal{X} \to \{\mathcal{H}_0, \mathcal{H}_1\}$ whose decision function is parametrized by the detection threshold $\tau \in \mathbb{R}$:

$$\delta(\mathbf{X}) = \begin{cases} \mathcal{H}_0 & if \quad \lambda(\mathbf{X}) \leq \tau, \\ \mathcal{H}_1 & if \quad \lambda(\mathbf{X}) > \tau \end{cases} . \tag{1}$$

The false alarm (or false positive) rate in subsequently defined as:

$$\mathbb{P}\left(\lambda(\mathbf{X}) > \tau \mid \mathcal{H}_0\right), \tag{2}$$

Where $\mathbb{P}(E)$ denotes the occurrence probability of event E.

On the opposite, the missed detection (or false negative) rate is defined as:

$$\mathbb{P}\left(\lambda(\mathbf{X}) \leq \tau \mid \mathcal{H}_1\right). \tag{3}$$

which is often measured with the "sensitivity" or power of the test:

$$\mathbb{P}\left(\lambda(\mathbf{X}) > \tau \mid \mathcal{H}_1\right). \tag{4}$$

Of course, as emphasized in Equations 2–4 both the false-positive and false-negative rates depend upon the decision threshold τ. In steganography, the performance of a steganalysis method is very often defined by the minimal total probability of error under equal prior, *i.e.* assuming covers and steganographic media as equally likely:

$$P_E = \min_{\tau \in \mathbb{R}} \frac{\mathbb{P}\left(\lambda(\mathbf{X}) > \tau | \mathcal{H}_0\right) + \mathbb{P}\left(\lambda(\mathbf{X}) \leq \tau | \mathcal{H}_1\right)}{2}. \tag{5}$$

Which is the opposite of the maximal accuracy one can get by adjusting the threshold.

The ensuing steganalysis problem of cover-source mismatch (CSM problem) is the degradation of a stego-detection performance when training and testing sets come from different cover-sources.

Table 1: An illustrative example showing the impact cover-mismatch problem: The diagonal contains intrinsic difficulties, and off-diagonal values are the inconsistencies ; Figure from our prior paper [32].

		Tested on source $\mathcal{S}_A$	Tested on source $\mathcal{S}_B$
Trained on	source $\mathcal{S}_A$	0.20 $\leftarrow P_E^{(\mathcal{S}_A)}$	0.38 $\leftarrow P_E^{(\mathcal{S}_A \to \mathcal{S}_B)}$
	source $\mathcal{S}_B$	0.33 $\leftarrow P_E^{(\mathcal{S}_B \to \mathcal{S}_A)}$	0.35 $\leftarrow P_E^{(\mathcal{S}_B)}$

Intrinsic difficulty — Inconsistency

For the sake of exemplification, let us consider two sources of images, , $\mathcal{S}_A$ and $\mathcal{S}_B$ respectively. We will denote $\mathbf{X} \sim \mathcal{S}_A$ (resp. $\mathbf{X} \sim \mathcal{S}_A$) when images come from the source $\mathcal{S}_A$ (resp. $\mathcal{S}_B$). Similarly, we will denote $\delta^{(A)}$ (resp. $\delta^{(B)}$) a detector that has been trained for detecting steganographic media coming from the source $\mathcal{S}_A$ (resp. $\mathcal{S}_B$).

The *intrinsic difficulty*, of source $\mathcal{S}_A$, measures how much it is "difficult" for the steganalyst to carry out the hidden information detection task on this source even when it is known. To this end, a usual approach consists in measuring the error rate for a given detector trained and tested on a source $\mathcal{S}_A$. In the present paper, as very often in media steganalysis, we use the usual total probability of error P_E as defined in (5) and will be denoted:

$$P_E^{(\mathcal{S}_A)} = \min_{\tau \in \mathbb{R}} \frac{\mathbb{P}^{(\mathcal{S}_A)}\left(\lambda(\mathbf{X}) > \tau | \mathcal{H}_0; \mathcal{S}_A\right) + \mathbb{P}^{(\mathcal{S}_A)}\left(\lambda(\mathbf{X}) \leq \tau | \mathcal{H}_1; \mathcal{S}_A\right)}{2}. \tag{6}$$

Here the superscripted notation $\mathbb{P}^{(\mathcal{S}_A)}$ means "trained over media from source $\mathcal{S}_A$".

The *intrinsic difficulty* is important as it serves as a baseline and represents steganalysis error without any mismatch. Indeed, as we shall see in the present paper, the intrinsic difficulty can vary significantly even for similar sources.

However, to characterize the cover-source mismatch one also needs to measure the detection error rate when facing a mismatch (i.e. when training and testing sources differ). To this end, let us define the *source inconsistency* which measures the error rate of a detector trained on source $\mathcal{S}_A$ and tested on source $\mathcal{S}_B$. In the present paper, we will use a measure based on the usual total probability of error P_E:

$$P_E^{(\mathcal{S}_A \to \mathcal{S}_B)} = \min_{\tau \in \mathbb{R}} \frac{\mathbb{P}^{(\mathcal{S}_A)}\left(\lambda(\mathbf{X}) > \tau | \mathcal{H}_0; \mathcal{S}_B\right) + \mathbb{P}^{(\mathcal{S}_A)}\left(\lambda(\mathbf{X}) \leq \tau | \mathcal{H}_1; \mathcal{S}_B\right)}{2} \tag{7}$$

It will be used in the present paper to assess the degradation of stego-detection performance in the presence of CSM and hence to assess CSM problem for AI generated sources, or *Deepfakes*.

Table 1 provides a toy example, in the very same manner as we shall present in the rest of this paper, with two sources only: rows represent the training source while columns represent the source

Table 2: Average of error rates (in %) of steganalysis carried out with EfficientNet-v2S (small size model) for the three main kinds of sources used in the present paper.

		Test		
		BOSS+BOWS	ALASKA2	*DeepFakes*
Train	BOSS+BOWS	13.99	37.49	44.56
	ALASKA2	37.90	26.35	35.29
	DeepFakes	40.03	35.98	23.27

used for testing. One can note that on the error rates on the diagonal represent the *intrinsic difficulty*. On the opposite, off-diagonal elements report the error rate in the presence when training over one source and testing over a different one hence assessing the *source inconsistency*. However, reporting the *source inconsistency* alone is insufficient; indeed it can only be interpreted by comparing the corresponding *intrinsic difficulty* of the testing source, hence reading the table column-wise. On the opposite, one row represent the "robustness capacity" of a cover-source.

2.2 Position of the present paper

Table 2 provides a motivating introductory example: (see Section 3.1 for detail about the experimental setup)
As explained, the images from BOSS+BOWS are homogeneous and contain a rather low noise level because of the harsh final rescaling to size 512×512. This can explain the much lower *intrinsic difficulty* over this dataset. However, we were also expecting a rather uniform and very limited noise level in deepfake images which would yield a low *intrinsic difficulty*. Surprisingly, one can note from Table 2 that this is not the case: indeed, those images seem as relevant for the steganographer as the ALASKA source which was designed to have a rather high, yet controlled, diversity for academic purposes. Similarly, one can note from Table 2 that the *source inconsistency* is extremely high in all cases but, as expected, slightly higher from images coming from deepfake sources.

These first results motivated the study carried out in the present paper and clearly explain both the position and the relevance of the present paper.
With the widespread of deepfakes, the ease with which one, without any knowledge, can use AI generators, and given the very fast pace at which the quality of such images has increased over the past few years, it is worth studying its applicability for hidden information detection. In the specific field of steganalysis, it is generally acknowledged that CSM constitutes one of the most fundamental issues for practical applications [26].
The present paper aims at studying the problem of steganalysis in deepfakes which, to the best of our knowledge, was never addressed. Our goal is to answer simple questions about the relevance of such sources of images for steganography, and the importance of CSM between AI generators.

We also explore possible solutions for the steganalyst. In this field, two main approaches have been used to expand simple detectors and partially mitigate the problem due to the CSM [32]. On the one hand, the *holistic approach* aims to train a single detector on diverse cover-sources, such that it enhances the generalization ability of the detector. Recent work such as [1] highlights the challenge in designing the training set. The *atomistic approach*, on the other hand, proposes a two-step detection: first a forensic tool identifies the cover-source of a given inspected image. Then, using one steganalysis detector per cover-source, the result of the forensics analysis is used to pick the most appropriate steganalysis detector. The main challenge lies in designing the forensic tool [33].
Those two approaches will be used in the present paper to assess the efficiency of these two usual mitigation strategies when cover-source are AI image generators.

3 EXPERIMENTAL METHODOLOGY AND RESULTS

3.1 Common core of all experiments

To assess the potential of deepfake images as covers for steganography, we carried out a large set of numerical experiments. For meaningful experimentations, three different spatial embedding algorithms from the state of the art, namely HILL [31], UNIWARD [21], and MiPOD. As shown in Table 3, we explore a considerable variety of deepfake generation methods, to encompass comprehensively

Table 3: List of generators used in our experiments. Their accessibility is given as links in the "Origin" column.

Type	Name	Origin	# of imgs
Diffusion Models	DeepFloyd-IF	Hugging Face	12,000
	Kandinsky v3	Hugging Face	14,000
	Pixart-α	Hugging Face	14,000
	Playground 2.5	Hugging Face	12,000
	Playground 2.0	Hugging Face	12,000
	DreamLike-PhotoReal 2.0	Hugging Face	12,000
	Stable-Diff 1.5	Hugging Face	12,000
	Stable-Diff 2.1	Hugging Face	12,000
	Stable-Diff XL	Hugging Face	12,000
	Animagine XL3.1	Hugging Face	12,000
	Total		122,000
GANs	GigaGAN	Github	10,000
	Glide	Github	12,000
	Dall•E Mini	Github	12,000
	StyleGAN 3	Github	18,000
	StyleGAN 2 ADA	Github	20,000
	Total		72,000
Testing sets	Dall-E 2	website	1,000
	Dall-E 3	website	1,000
	Midjourney v5	website	1,000
	Adobe Firefly	website	1,000
	Stable-Diff 1.3	Hugging Face	1,000
	Stable-Diff 1.4	Hugging Face	1,000
	Stable-Diff 2.0	Hugging Face	1,000
	Stable-Diff XL	Hugging Face	1,000
	Total		8,000
photo	BOSS [3] + BOWS [4]	Link	20,000
	ALASKA2 [10]	website	80,000
	Total		100,000

Table 4: Total Error rates, P_E (5) in %, for both detectors under assumption that the cover-source is known.

	DF	Kand	PixArt	PG2.5	PG2	DL-PR	SD1.5	SD2.1	SD-XL	Anim	megaDallE	Giga	Glide	SGan2	SGan3
SRMQ1 + LC	16.68	27.40	24.93	16.54	8.07	22.95	26.66	23.37	29.40	0.91	21.42	10.30	4.42	18.42	3.17
EfficientNet-v2	16.86	32.15	26.73	19.06	8.20	29.22	30.30	25.35	41.96	0.62	26.68	8.29	9.82	17.71	3.47

Table 5: Comparison of average change rates, in %, ratio of the expected number of hidden bits divided by the total number of pixels, for different cover-source and various embedding algorithms. For comparison the P_E are the one from Table 4.

	HILL	S-UNIW	MiPOD	P_E
ALASKA	12.48	11.49	12.49	26.35
mix/all	14.79	12.33	15.92	23.27
Kandinsky	15.93	13.13	16.68	27.40
PlayGround 2	15.01	12.30	16.44	8.07
DreamLike-PR	13.39	12.20	14.21	22.95
Stable-Diff 1.5	13.71	12.31	14.04	30.30
Stable-Diff XL	14.27	12.22	14.66	41.96
Animagine 3.1	17.41	12.99	18.31	0.62
GigaGAN	13.99	12.15	15.54	8.29
Glide	16.12	12.20	16.14	9.82
StyleGAN 2	12.95	11.81	13.99	17.71
StyleGAN 3	12.57	11.40	13.34	3.47

the current existing art, from first GAN models, *e.g.* StyleGAN 2 [23], StyleGAN 3 [24], GigaGAN [22], up to the latest diffusion models, namely StableDiffusion 2.1, XL [35], Kandinsky [36] and Pixart-α [6] to cite a few. For comparison, we also used two reference datasets of images for digital image steganography. On the one hand, we have used BOSS [3] and BOWS [4] bases combined, both made of $10,000$ grayscale images of size 512×512. As explained in [9, 16] this dataset is very specific because all images have been processed in the same way and especially largely resized. This dataset is therefore expected to exhibit results consistent with that uniformity. On the other, to compare the security of embedding in deepfake images with more realistic photograph datasets, we have also used the recent ALASKA base [9, 10] which is made of $80,000$ grayscale images of size 512×512. Every image from this dataset has been processed differently using a randomized process.

Note that Deepfake image generators mostly produce color uncompressed images. We convert them into grayscale (retaining only the luminance channel Y) as steganography has been very seldom explored in color images (see for instance [11, 12, 20]).

We used the three main state-of-the-art embedding algorithms for steganography, namely S-UNIWARD [21], HILL [31] and MiPOD [37]. However, due to space limitations, we present in this paper results for the two formers only.

3.2 Experimental method for assessment of CSM problem with deepfakes

We adopted the empirical assessment approach [16, 33], that is, we measure the impact of the CSM between deepfake sources, we opt for the formulas given in Eq. 6 for the intrinsic difficulty and Eq. 7 for the source inconsistency.

To this end, we used two main steganalysis methods to assess the detectability of steganography over the different sources of images. On the one hand, we used the most established features-based method, namely SRMQ1 features set [19] with the fast-linear classifier [13]. On the other, we also included results from steganalysis based on deep learning because they now constitute the state-of-the-art. We used the recent and already well-established EfficientNet-v2 [39, 40], as it has been shown to be extremely efficient for steganalysis during the ALASKA Steganalysis Challenge [10], see for instance [7, 43].

To train this deep learning-based classifier, we used two simple yet important tricks to speed up the convergence process. First, it has been shown during the ALASKA Challenge that, even though the classification task is very different from steganalysis, using weights pre-trained from Imagenet dramatically speeds up convergence. To this end, we used the `timm` Python package for `pytorch`, which offers pre-trained models of a wide range of popular deep-learning architectures. Second, we also adopted a curriculum learning very similar to the one proposed in [42], starting with non-adaptive LSBM steganography, reducing the payload step by step, and then retraining the model on the different embedding methods starting again from higher payload. During this curriculum training phase, we applied the following setting at each step: we used 15 epochs starting with a Learning Rate of 10^{-3}, reducing this hyperparameter slowly with the "reduced on plateau" strategy. During the "fine-tuning" stage, we used 25 epochs starting with the maximal Learning Rate of 10^{-4} and applied the cosine annealing strategy with a minimal learning rate of 10^{-7}.

4 RESULTS AND ANALYSIS

4.1 Results on the Relevance of DeepFake Sources: Intrinsic Difficulty

First, let us contrast the *intrinsic difficulty* for all the sources of AI image generation presented in Table 3. For clarity, we recall that the intrinsic difficulty (6) is measured as the P_E (5) in the absence of a mismatch, that is when training and testing the steganalysis over the same AI generator.

Note we used a very large dataset of over $200,000$ AI generated images, see the Table 3, and mainly consists of images we generated ourselves, via the `Hugging Face` library or Github repositories with available pre-trained weights. For the proprietary generators, such as Adobe's Firefly, samples were gathered previously available datasets and, due to their much smaller number of images, these will only be used during the testing phase. Table 4[2] reports the P_E (5) obtained with the two aforementioned detectors over the 15

[2]the following abbreviations are used in column headers : DF: DeepFloyd ; Kand: Kandinsky v3 ; PixArt: Pixart-α PG: Playground DL-PR: DreamLike-PhotoReal v2.0 ;

Table 6: Total Error rates, P_E (5) in %, of steganalysis using EfficientNet-v2 (small) according to the training and testing datasets. The diagonal elements show the supposedly minimal error rate (when training and testing sets match) while each column shows when training with the wrong AI deep fake generator.

		testing dataset														
		DF	Kand	PixArt	PG2.5	PG2	DL-PR	SD1.5	SD2.1	SD-XL	Anim	DallE	Giga	Glide	SGan2	SGan3
training dataset	Deep Floyd	16.86	47.62	41.57	40.17	27.26	45.92	48.50	36.05	46.91	28.97	45.04	22.12	36.51	32.45	15.65
	Kandinsky	37.67	32.15	44.71	44.38	39.96	39.50	41.92	42.83	41.54	18.31	40.09	34.31	25.56	40.98	45.72
	PixArt	31.22	45.14	26.73	27.89	11.99	42.42	46.87	34.59	43.00	2.83	38.09	22.12	25.52	31.05	21.48
	PlayGround 2.5	34.43	47.65	38.33	19.06	12.19	42.17	47.50	38.88	39.00	2.37	44.87	19.83	38.51	33.33	33.06
	PlayGround 2	31.89	48.51	43.82	31.55	8.20	46.66	47.12	40.30	38.88	2.45	46.42	18.98	42.63	35.23	24.65
	DreamLike-PR	36.26	43.90	39.79	35.88	27.60	29.22	38.55	41.50	44.21	3.78	41.59	26.62	32.39	36.30	34.12
	StableDiff 1.5	37.05	39.52	42.89	38.67	32.01	39.59	30.30	43.83	41.67	4.78	39.63	36.86	26.02	35.63	31.56
	StableDiff 2.1	25.06	43.10	37.25	33.13	13.98	43.00	45.62	25.35	39.75	7.32	33.22	34.96	18.81	29.76	24.59
	StableDiff XL	40.34	43.24	39.61	36.13	28.01	41.34	43.29	41.04	41.96	12.24	41.84	40.01	33.59	40.73	36.42
	Animagine 3.1	32.26	45.38	40.04	32.39	16.98	41.25	41.71	40.17	40.75	0.62	40.92	20.62	41.00	30.78	13.02
	megaDallE	30.55	43.38	40.79	37.51	18.19	42.67	44.33	31.76	41.04	18.60	26.68	30.81	15.69	31.48	32.14
	GigaGAN	32.76	49.27	40.25	42.29	18.77	47.25	46.54	38.46	48.25	11.61	46.91	8.29	36.71	31.45	9.30
	Glide	28.72	45.69	35.54	31.55	16.77	43.75	44.71	32.14	43.79	7.57	31.84	21.62	9.82	30.58	14.43
	StyleGAN v2	35.55	46.41	44.43	37.21	31.59	40.67	42.29	44.08	45.33	9.74	41.21	37.31	27.10	17.71	31.84
	StyleGAN v3	35.09	49.41	46.21	49.00	37.51	49.79	49.79	45.29	49.33	41.29	49.50	45.70	42.88	40.98	3.47
	ALASKA	28.31	47.41	39.36	42.42	21.27	42.38	44.96	38.96	45.92	12.57	46.62	19.68	37.09	33.90	7.96
	holistic	20.19	36.35	30.22	23.18	10.28	33.59	34.97	30.01	34.43	0.99	32.01	12.43	12.32	21.01	5.60
	atomistic	16.91	32.31	26.74	19.08	8.16	29.08	30.37	25.25	41.54	0.58	26.70	8.78	9.83	17.67	3.44

Table 7: Total Error rates, P_E (5) in %, of steganalysis using SRMQ1 + Linear Classifier according to the training and testing datasets. The diagonal elements show the supposedly minimal error rate (when training and testing sets match) while each column shows when training with the wrong AI deep fake generator.

		testing dataset														
		DF	Kand	PixArt	PG2.5	PG2	DL-PR	SD1.5	SD2.1	SD-XL	Anim	DallE	Giga	Glide	SGan2	SGan3
training dataset	Deep Floyd	16.68	45.27	44.30	40.82	36.90	48.62	46.75	48.22	48.13	17.91	42.52	46.62	36.34	42.88	26.73
	Kandinsky	37.27	27.40	41.78	45.49	31.98	44.07	44.94	43.18	41.57	9.71	40.15	39.27	21.50	44.27	38.70
	PixArt	43.45	45.07	24.93	40.06	29.45	48.18	45.95	42.20	47.30	9.25	42.40	36.54	20.47	48.96	39.88
	PlayGround 2.5	34.87	45.87	44.31	16.54	17.58	49.16	49.67	47.99	41.52	31.63	44.15	43.30	30.62	47.09	33.36
	PlayGround 2	38.25	47.72	48.12	30.88	8.07	49.96	49.93	47.40	44.99	8.80	46.20	46.56	37.07	43.36	26.35
	DreamLike-PR	39.08	43.28	43.59	40.83	40.85	22.95	43.83	41.87	47.98	13.90	39.06	35.96	29.93	41.74	39.54
	StableDiff 1.5	34.22	41.47	39.57	45.25	32.97	37.77	26.66	42.77	45.15	21.05	37.42	36.14	30.31	38.68	35.00
	StableDiff 2.1	39.67	42.03	36.40	42.68	24.00	44.54	45.44	23.37	44.30	18.04	32.37	36.82	20.39	39.58	29.37
	StableDiff XL	46.27	45.52	44.80	30.08	21.13	49.22	49.61	46.05	29.40	18.42	40.94	43.24	24.71	43.98	36.94
	Animagine 3.1	36.53	41.70	41.95	36.54	33.44	49.04	47.14	48.07	46.54	0.91	39.38	39.40	22.05	43.39	37.36
	megaDallE	36.10	40.93	45.11	45.63	36.59	44.64	47.07	39.64	41.50	29.21	21.42	37.90	14.92	44.70	32.36
	GigaGAN	42.07	49.31	46.19	45.63	29.83	49.99	49.79	46.57	49.73	33.68	48.94	10.30	37.64	45.42	32.47
	Glide	32.11	41.65	36.14	41.26	23.05	47.13	46.41	36.09	41.48	7.49	28.62	37.26	4.42	40.93	30.72
	StyleGAN v2	48.40	50.01	47.50	43.89	49.77	49.71	47.65	48.27	49.82	38.46	44.27	34.35	41.77	18.42	40.01
	StyleGAN v3	41.55	49.43	48.51	48.31	42.88	49.98	49.74	49.25	49.82	24.52	49.17	48.02	43.31	39.94	3.17
	holistic	24.59	37.13	35.14	33.96	18.30	35.74	38.02	33.65	38.53	1.44	30.35	23.28	10.33	25.66	12.54

AI text-to-image generators. To keep things concise, we report the intrinsic difficulties for HILL embedding with the SRMQ1 feature-based detector (first row) and MiPOD embedding scheme with the EfficientNet-v2 steganalyzer (second row). Several interesting

Anim: Animagine v3.1 ; DallE: mageDallE ; Giga: GigaGAN and SGan2: StyleGan 2 ADA and SGan3: StyleGan 3 see Table 3 for details.

Table 8: Analysis of robustness with respect to changes in the deepfakes generation process via the total Error rates, P_E (5) in %, of steganalyzers using EfficientNet-v2 (small) trained on our own dataset and test on other datasets

		testing dataset							
		D·2	D·3	FF	MJ-5	SD1.3	SD1.4	SD2	SD-XL
training dataset	Deep Floyd	49.7	46.9	29.2	45.7	49.0	49.3	48.7	47.8
	Kandinsky	49.6	47.1	46.8	45.6	48.1	48.1	46.5	44.4
	PixArt	49.4	46.1	36.8	43.1	49.1	49.1	47.2	39.1
	PlayGround 2.5	47.9	47.1	32.8	46.4	49.0	49.0	47.9	35.5
	PlayGround 2	49.4	48.0	34.3	46.6	49.2	49.5	47.8	39.6
	DreamLike-PR	49.7	46.0	40.8	46.0	47.1	47.6	46.5	39.2
	StableDiff 1.5	49.1	46.9	41.5	45.5	46.7	46.1	46.1	43.2
	StableDiff 2.1	49.5	47.2	39.8	39.6	48.6	48.7	41.3	37.8
	StableDiff XL	49.3	46.6	43.7	45.3	48.2	47.6	46.9	41.3
	Animagine 3.1	48.6	46.7	32.7	45.0	48.4	48.1	47.5	41.9
	megaDallE	49.7	46.5	44.9	38.6	48.6	48.7	43.3	38.6
	GigaGAN	49.5	46.4	40.4	46.8	48.8	49.0	48.2	48.5
	Glide	49.5	46.2	34.2	40.5	49.0	49.1	44.6	41.8
	StyleGAN v2	46.8	45.4	44.0	45.3	47.3	47.6	46.6	41.7
	StyleGAN v3	49.8	49.0	28.5	48.1	49.6	49.7	49.5	49.8
	mix/all	49.7	45.2	33.5	38.9	46.6	47.1	44.7	36.2
	ALASKA	46.5	42.3	24.9	46.1	48.3	48.4	48.0	42.5

conclusions can be made from these results.
First of all, the *intrinsic difficulty* varies dramatically between the different generators, ranging from less than 1% from Animagine-XL-3.1 to almost 30% for StableDiffusion-XL and Kandinsky, in the case of steganalysis with SRMQ1 and LCLC, and even more than 40% with EfficientNet-v2. Similarly, it can be noted that some generators whose architectures are very close present very different *intrinsic difficulties*. See for instance the couple's Animagine-XL-3.1 which is a fine-tuning of StableDiffusion-XL, PlayGround-v2.5 which derives from PlayGround-v2 and StyleGAN-v2 which derives StyleGan-v3.
Second, the use of SRMQ1 features with the low-complexity linear classifier (LCLC) is surprisingly efficient, even slightly better then EfficientNet. This can be explained in part by the fact that the number of images for each AI text-to-image generator is rather limited to train a rather complex deep learning detector such as EfficientNet-v2.

Table 9: Overall comparison of the robustness of detectors trained over the two main kinds of deepfakes generators: GANs and Diffusion models. Average P_E (5), in %, over all other generators (excluding the only matching case).

		Tested on	
		GANs	Diffusion models
Trained on	GANs	40.0	45.4
	Diffusion models	39.2	38.2

Eventually, we explored the average changing rate, that is the number of actual pixels modified (in our case to allow the embedding of 0.4 bpp) over these different cover-sources. Indeed, this information is important as it tells meaningful information about the "distribution of content complexity" and its relevance with respect to a steganographic scheme. For instance, an overly adapted steganographic algorithm will tend to concentrate the payload in textured areas, reducing the embedding efficiency and hence the higher changing rate. On the opposite, smoothed image content and uniform noise yield similar costs over all pixels, increasing the embedding efficiency and hence the lower changing rate.
These results are reported in Table 5 which, for the sake of comprehensiveness also reports the average over the ALASKA dataset along with the intrinsic difficulty. One can observe that the embedding efficiency is almost always lower for AI-generated images than for ALASKA images. An exception is S-UNIWARD with Style Gan 3, which provides a ratio of 11.40%, slightly better than that of ALASKA (11.49%). This can be explained by the fact that the steganographic methods are designed for natural photography. However, this also points out the fact that image steganography is more "adaptive" over deepfake images. The second interesting result is that it does not seem that adaptivity to image source content plays a significant role. However, the sources with the lowest difficulty are those for which the change rate is the highest, hence the embedding into more restrictive areas, see for instance the comparison between StyleGan-v2 and StyleGan-v3.

4.2 Results on the Robustness of steganalysis: Inconsistency Between Sources

Let us now move on to the study of the cover-source mismatch between AI text-to-image generators. To this end, we have adopted an experimental method that consists of testing all classifiers, hence trained for a specific cover-source, over each and every dataset of images from a different source. As explained in Section 2.1, this provides us with the inconsistency and should be compared with the intrinsic difficulty of the testing dataset in order to assess the "robustness" of steganalyzers with respect to the cover-source hence the importance of the CSM problem.

Table 7 and 6. provide error rates of steganalysis using the SRMQ1 feature-based classifier and EfficentNet, respectively. Several interesting conclusions can be drawn from these tables. First and foremost, one can note that deepfake sources exhibit high inconsistency between them: the average intrinsic difficulty, given in Table 4 is about 19.5% (resp. 17%) for SRMQ1 (resp. EfficientNet), the average inconsistency increases to 38% (resp. 35%). More surprisingly, cover-sources supposedly close to each other do not necessarily yield lower inconsistencies: for example, Animagine XL has an intrinsic difficulty lower than 1% whereas the inconsistency when training over StableDiffusion-XL goes up to 15% while the former is a fine-tuned version of the latter. Similar observations can be made for DreamLike-PhotoReal and StableDiffusion-v1.5, StableDiffusion-XL and Playground-2.0, or Stylegan-v2 and Stylegan-v3.
Note that these important inconsistencies do not stem from different prompts; indeed, we used the same prompts for every generator;

therefore, high inconsistencies can only be explained by the deepfake cover-sources.
We further confirmed these observations using an additional dataset from [2]. Table 8[3]. shows a selective subset of inconsistencies obtained when training on the same 15 generators as in Table 7 and 6, but testing on these additional images, as for Table 6. Steganalysis is performed with EfficientNet-v2 against the MiPOD embedding algorithm with payload 0.4 bpp (Bits Per Pixels).
Similar conclusions can be drawn from [2]: the inconsistency between deepfake cover-sources is very important even for models supposedly close to each other (see the family of StableDiffusion for instance). However, one can observe that the only common cover-source, namely StableDiffusion-XL, exhibits similar results as the one reported in Table 6. This seems to point out that the specific hyperparameter and overall settings of the AI image generator (prompt, diffusion steps, etc.) may have a limited impact on cover-source mismatch.

Interestingly, it appears that none of the cover-source allows mitigation of the cover-source mismatch problem. However one can note that, generally speaking, diffusion models tend to offer a slightly higher robustness, hence the slightly lower inconsistency. To better highlight this phenomenon, Table 9 presents the average inconsistency over all GAN image generators (megaDallE, GigaGAN, Glide, StyleGAN v2 and v3) and Diffusion models. Even when testing on deepfakes generated from GAN-based image generators, the average inconsistency is lower when training on diffusion models.

Last but not least, we have tried implementing two usual strategies for mitigating the cover-source mismatch: namely the holistic and the atomistic approaches.
The holistic approach, which consists of training a steganalysis classifier over a dataset made from all the sources merged together, provides overall quite satisfactory results with an average error rate P_E of 22.5% and 26.5% for EfficientNet and SRMQ1 respectively. While this result is a significant improvement to the very high inconsistencies presented in Table 6 and 7 averaging to 35% and 38%.
However, this must be tempered by comparing it with the intrinsic difficulties from Table 4 of about 17% and 19.5% respectively.

On the opposite the atomistic approach consists in a two-step steganalysis process: first a multiclass classifier method is trained to identify which AI generator a given deepfake image has been produced with. Then, the second step is made of as many binary steganalysis classifier as there are different deepfake generators. Therefore, when an atomistic detector is given an image to inspect it tries to identify, first, the possible AI generator and then it applies the steganalysis detector trained specifically for the most likely cover-source. This approach seems extremely accurate for EfficientNet, see the Table 6, but this results must be confirmed by replications in larger-scale and diverse experiments.

[3]the following abbreviations are used in column headers : D·2: Dalle·2 ; D·3: Dalle·3 ; FF: Adobe FireFly ; MJ-5: MidJourney v5 and SD: StableDiffusion ; see Table 3 for details.

5 CONCLUSIONS

The use of AI-generated images, so-called *deepfakes*, has been skyrocketing and such technology has become widely accessible off-the-shelf over the past few years . The potential misuse of such technology, such as spreading misinformation, impersonating individuals, or even forging evidence, has been broadly studied.

On the opposite, the present paper anticipates the case when such images will often be exchanged and study their applicability to steganography. To the best of our knowledge, such an evaluation has never been proposed. Using an empirical method and state-of-the-art tools, we have assessed how relevant such a source of images can be used to hide sensitive data. We have also studied how much one could leverage the Cover-Source Mismatch problem in steganalysis, which is widely acknowledged as a fundamental barrier for steganalysis in an operational context, using the variability of existing AI text-to-image generators.
Surprisingly, the experimental results presented in this paper show that *deepfakes* can provide a source of images that is rather difficult to inspect for steganalysis. However, steganographers should use it with caution as this highly depends on the AI image generator model.
In addition, our results show that it is very difficult for the steganalyst to inspect *deepfakes* images regardless of the models used as a generator. Eventually, while the steganalyst does have some methods to mitigate the cover-source mismatch problem, our results seem to indicate that both holistic and atomistic approaches have some limitations and yield substantial loss of steganalysis detection accuracy.

The present paper unveils a particularly interesting blind spot of AI image generation. However, we shall also emphasize that some future works are required to confirm and generalize these results, and to explain them in greater detail. We especially have in mind studies to explain which characteristics of deepfake images can explain such a difference in terms of steganalysis intrinsic difficulty. Similarly, the sensitivity of steganalysis with respect to AI image generator parameters, their content, their variability, their noise level, etc. is a crucial factor that shall be investigated to understand and explain better the reasons behind the tremendous cover-source mismatch effect we report.
Last, but not least, experiments with JPEG steganography in order to study on the effect of JPEG compression on the very high CSM over AI generated images is an interesting future work.

ACKNOWLEDGMENTS

This work has been funded by the EU's Horizon 2020 program under grant agreement No. 101021687(UNCOVER), and the French ANR PACeS project No. ANR-21-CE39-0002.

REFERENCES

[1] Rony Abecidan, Vincent Itier, Jérémie Boulanger, Patrick Bas, and Tomáš Pevný. 2022. Using Set Covering to Generate Databases for Holistic Steganalysis. In *WIFS*. IEEE, 1–6.
[2] Quentin Bammey. 2024. Synthbuster: Towards Detection of Diffusion Model Generated Images. *IEEE Open Journal of Signal Processing* 5 (2024), 1–9. https://doi.org/10.1109/OJSP.2023.3337714
[3] P. Bas, T. Filler, and T. Pevný. 2011. Break Our Steganographic System — the ins and outs of organizing BOSS. In *Information Hiding, 13th International Workshop (Lecture Notes in Computer Science)*. LNCS vol.6958, Springer-Verlag, New York, Prague, Czech Republic, 59–70. agents.fel.cvut.cz/boss/

[4] Patrick Bas and Teddy Furon. July 2007. BOWS-2 Contest (Break Our Watermarking System). http://bows2.ec-lille.fr/

[5] Rainer Böhme. 2010. *Advanced Statistical Steganalysis* (1st ed.). Springer Publishing Company, Incorporated.

[6] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. 2023. PixArt-α: Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis. *arXiv preprint arXiv:2310.00426* (2023).

[7] Kaizaburo Chubachi. 2020. An Ensemble Model using CNNs on Different Domains for ALASKA2 Image Steganalysis. In *Information Forensics and Security (WIFS), IEEE 12th International Workshop on* (New York, NY, USA). 4.

[8] Rémi Cogranne and Jessica Fridrich. 2015. Modeling and Extending the Ensemble Classifier for Steganalysis of Digital Images Using Hypothesis Testing Theory. *Information Forensics and Security, IEEE Transactions on* 10, 12 (2015), 2627–2642. https://doi.org/10.1109/TIFS.2015.2470220

[9] Rémi Cogranne, Éva Giboulot, and Patrick Bas. 2019. The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security* (Paris, France) *(IH&MMSec'19)*. ACM, New York, NY, USA, 125–137. https://doi.org/10.1145/3335203.3335726

[10] Rémi Cogranne, Éva Giboulot, and Patrick Bas. 2020. ALASKAv2: Challenging Academic Research on Steganalysis with Realistic Images. In *Information Forensics and Security (WIFS), IEEE 12th International Workshop on* (New York, NY, USA). 4.

[11] Rémi Cogranne, Éva Giboulot, and Patrick Bas. 2020. Steganography by Minimizing Statistical Detectability: The Cases of JPEG and Color Images. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security* (Denver, CO, USA) *(IH&MMSec'20)*. Association for Computing Machinery, New York, NY, USA, 161–167. https://doi.org/10.1145/3369412.3395075

[12] Rémi Cogranne, Eva Giboulot, and Patrick Bas. 2022. Efficient Steganography in JPEG Images by Minimizing Performance of Optimal Detector. *IEEE Transactions on Information Forensics and Security* 17 (2022), 1328–1343. https://doi.org/10.1109/TIFS.2021.3111713

[13] Rémi Cogranne, Vahid Sedighi, Jessica Fridrich, and Tomáš Pevný. 2015. Is Ensemble Classifier Needed for Steganalysis in High-Dimensional Feature Spaces?. In *Information Forensics and Security (WIFS), IEEE 7th International Workshop on* (Rome, Italy). 1–6.

[14] Tomáš Denemark, Vahid Sedighi, Vojtěch Holub, Rémi Cogranne, and Jessica Fridrich. 2014. Selection-Channel-Aware Rich Model for Steganalysis of Digital Images. In *Information Forensics and Security (WIFS), IEEE 6th International Workshop on* (Atlanta, GA, USA). 48–53.

[15] Éva Giboulot, Rémi Cogranne, and Patrick Bas. 2018. Steganalysis into the Wild: How to Define a Source?. In *Media Watermarking, Security, and Forensics (Proc. IS&T)*. 318–1 – 318–12. https://doi.org/10.2352/ISSN.2470-1173.2018.07.MWSF-318

[16] Éva Giboulot, Rémi Cogranne, Dirk Borghys, and Patrick Bas. 2020. Effects and solutions of Cover-Source Mismatch in image steganalysis. *Signal Processing: Image Communication* 86 (2020), 115888. https://doi.org/10.1016/j.image.2020.115888

[17] T. Filler, J. Judas, and J. Fridrich. 2011. Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes. *Information Forensics and Security, IEEE Transactions on* 6, 3 (Sept 2011), 920–935. https://doi.org/10.1109/TIFS.2011.2134094

[18] Jessica Fridrich. 2009. *Steganography in Digital Media: Principles, Algorithms, and Applications* (1st edition ed.). Cambridge University Press.

[19] J. Fridrich and J. Kodovskỳ. 2012. Rich Models for Steganalysis of Digital Images. *Information Forensics and Security, IEEE Transactions on* 7, 3 (june 2012), 868 –882. https://doi.org/10.1109/TIFS.2012.2190402

[20] Miroslav Goljan, Jessica Fridrich, and Rémi Cogranne. 2014. Rich model for Steganalysis of color images. In *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*. 185–190. https://doi.org/10.1109/WIFS.2014.7084325

[21] Vojtěch Holub, Jessica Fridrich, and Tomàš Denemark. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security* 2014, 1 (2014), 1–13. https://doi.org/10.1186/1687-417X-2014-1

[22] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. 2023. Scaling up GANs for Text-to-Image Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[23] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020. Training generative adversarial networks with limited data. *Advances in neural information processing systems* 33 (2020), 12104–12114.

[24] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. *Advances in neural information processing systems* 34 (2021), 852–863.

[25] Andrew Ker. 2005. Steganalysis of LSB Matching in Grayscale Images. *Signal Processing Letters* 12, 6 (2005), 441–444.

[26] Andrew Ker, Patrick Bas, Rainer Böhme, Rémi Cogranne, Scott Craver, Tomáš Filler, Jessica Fridrich, and Tomáš Pevný. 2013. Moving Steganography and Steganalysis from the Laboratory into the Real World. In *IH&MMSec.* ACM, 45–58.

[27] Mehdi Kharrazi, Husrev Sencar, and Nasir Memon. 2005. Benchmarking Steganographic and Steganalysis Techniques. In *SSWMC*, Vol. 5681. SPIE, 252–263.

[28] J. Kodovskỳ, J. Fridrich, and V. Holub. 2012. Ensemble Classifiers for Steganalysis of Digital Media. *Information Forensics and Security, IEEE Transactions on* 7, 2 (April 2012), 432–444. https://doi.org/10.1109/TIFS.2011.2175919

[29] Jan Kodovský, Vahid Sedighi, and Jessica Fridrich. 2014. Study of cover source mismatch in steganalysis and ways to mitigate its impact. In *Media Watermarking, Security, and Forensics (Proc. SPIE, Vol. 9028)*. Article 90280J, 90280J pages. https://doi.org/10.1117/12.2039693

[30] Vaila Leask, Rémi Cogranne, Dirk Borghys, and Helena Bruyninckx. 2022. UNCOVER: Development of an efficient steganalysis framework for uncovering hidden data in digital media. In *ARES 2022: The 17th International Conference on Availability, Reliability and Security, Vienna,Austria, August 23 - 26, 2022*. ACM, 44:1–44:8. https://doi.org/10.1145/3538969.3544468

[31] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li. 2014. A new cost function for spatial image steganography. In *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 4206–4210.

[32] Antoine Mallet, Martin Benes, and Rémi Cogranne. 2024. Cover-Source Mismatch in Steganalysis: Systematic Review. Under Review (2024). https://doi.org/10.2352/ISSN.2470-1173.2016.8.MWSF-076

[33] Antoine Mallet, Rémi Cogranne, and Patrick Bas. 2024. Statistical Correlation as a Forensic Feature to Mitigate the Cover-Source Mismatch. In *International Conference in Information Hiding and Multimedia Security (IH&MMSec)*. ACM, *In Press*.

[34] Tomás Pevný, Tomás Filler, and Patrick Bas. 2010. Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. In *Information Hiding*, Rainer Böhme, Philip Fong, and Reihaneh Safavi-Naini (Eds.). Lecture Notes in Computer Science, Vol. 6387. Springer Berlin / Heidelberg, 161–177.

[35] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952* (2023).

[36] Anton Razzhigaev, Arseniy Shakhmatov, Anastasia Maltseva, Vladimir Arkhipkin, Igor Pavlov, Ilya Ryabov, Angelina Kuts, Alexander Panchenko, Andrey Kuznetsov, and Denis Dimitrov. 2023. Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion. *arXiv preprint arXiv:2310.03502* (2023).

[37] V. Sedighi, R. Cogranne, and J. Fridrich. 2016. Content-Adaptive Steganography by Minimizing Statistical Detectability. *IEEE Transactions on Information Forensics and Security* 11, 2 (Feb 2016), 221–234. https://doi.org/10.1109/TIFS.2015.2486744

[38] Vahid Sedighi, Jessica J. Fridrich, and Rémi Cogranne. 2016. Toss that BOSSbase, Alice!. In *Media Watermarking, Security, and Forensics (Proc. IS&T)*. pp. 1–9. https://doi.org/10.2352/ISSN.2470-1173.2016.8.MWSF-076

[39] Mingxing Tan and Quoc Le. 2021. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*. PMLR, 10096–10106.

[40] Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proc. of Intl' Conference on Machine Learning, ICML 2019* (Long Beach, California, USA), Vol. 97. PMLR, 6105–6114. https://icml.cc/Conferences/2019/ScheduleMultitrack?event=3661

[41] Andreas Westfeld. 2001. F5–A Steganographic Algorithm. In *Information Hiding*. Lecture Notes in Computer Science, Vol. 2137. Springer Berlin / Heidelberg, 289–302.

[42] Yassine Yousfi, Jan Butora, Jessica Fridrich, and Éva Giboulot. 2019. Breaking ALASKA: Color Separation for Steganalysis in JPEG Domain. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security* (Paris, France) *(IH&MMSec'19)*. Association for Computing Machinery, New York, NY, USA, 138–149. https://doi.org/10.1145/3335203.3335727

[43] Yassine Yousfi, Jan Butora, Eugene Khvedchenya, and Jessica Fridrich. 2020. ImageNet Pre-trained CNNs for JPEG Steganalysis. In *Information Forensics and Security (WIFS), IEEE 12th International Workshop on* (New York, NY, USA). 4.