The Cover Source Mismatch Problem in Deep-Learning Steganalysis

Giboulot Quentin
LIST3N
University of Technology of Troyes
Troyes, France
quentin.giboulot@utt.fr

Bas Patrick
Univ. Lille, CNRS,
Centrale Lille
UMR 9189 CRIStAL
F-59000 Lille, France
patrick.bas@univ-lille.fr

Cogranne Rémi
LIST3N
University of Technology of Troyes
Troyes, France
remi.cogranne@utt.fr

Borghys Dirk
Royal Military Academy
Dept. Mathematics
Brussels, Belgium
Dirk.Borghys@rma.ac.be

***Abstract*—This paper studies the problem of Cover Source Mismatch (CSM) in steganalysis, i.e. the impact of a testing set which does not originate from the same source than the training set. In this study, the trained steganalyzer uses state of the art deep-learning architecture prone to better generalization than feature-based steganalysis. Different sources such as the sensor model, the ISO sensitivity, the processing pipeline and the content, are investigated. Our conclusions are that, on one hand, deep learning steganalysis is still very sensitive to the CSM, on the other hand, the holistic strategy leverages the good generalization properties of deep learning to reduce the CSM with a relatively small number of training samples.**

I. Introduction and Prior Works

Steganalysis is the discipline concerned with the detection of hidden data in innocuous cover media. Its history in the specific domain of natural images has been marked by tremendous advances in detection performance since the introduction of the supervised learning framework. The trend started with the combination of the SPAM feature set and the Support Vector machine in [1]. It continued with the undisputed success of the Spatial Rich models combined with Ensemble classifier during the BOSS competition [2]. This method was soon adapted to the JPEG domain with feature sets such as DCTR [3] and GFR [4]. The years 2015-2016 marked another step with the introduction of deep neural network which quickly showed their supremacy in the discipline with, in chronological order, Xu-Net [5], [6], SRNet [7] and most recently the use of EfficientNet [8] combined with transfer learning.

However, this steady increase of the performance of detectors in steganalysis should be contrasted with the relative stagnation of the studies on the applicability of these approaches to real world situations. Indeed, as stated at the beginning, these approaches all rely on the classical supervised learning framework. Therefore, they also inherit its limits, in particular, the lack of generalization guarantees on unseen data.

In a real world scenario, the steganalyst wants to classify one or several images as either cover or stego images. She should do so knowing that, while a missed detection is acceptable, a false alarm is very costly. Ideally then, she should be able to control the probability of false alarm of her detector and try to minimize the probability of missed detection. Sadly, the empirical performance computed with supervised learning methods are only guaranteed to generalize if the unseen data has been generated by using the same probability distribution as the training data. The mismatch between the training data distribution and the unseen data, usually termed cover-source mismatch in steganalysis, has been known to lead to a substantial loss of performance.

This phenomenon of cover-source mismatch was first documented in [9] where it was observed that training a classifier on a dataset containing images only taken with a given camera *CAM1* and testing it on a second dataset built only using another camera *CAM2* led to far worse performance than when the classifier was tested on a dataset built only with *CAM1*. This issue became even more evident during the BOSS competition where the organizers added images in the testing set which were taken with a camera not present in the training set. This induced a large drop in steganalysis performance on these very images. What is often less highlighted is that these outliers were not only taken with a unknown camera, but that they had all followed a double JPEG compression contrary to the other images which were simply compressed once. This shows that the processing pipeline might also play an important role on steganalysis performance.

The work in [10] studies this phenomenon by focusing mainly on the impact of different cameras, though it incidentally shows the greater impact of different processing pipelines on the effects of cover-source mismatch. The same authors also studied the effect of different resizing algorithms on steganographic security, demonstrating the key role of the processing pipeline.

Following the two ALASKA competitions [11], [12], deep neural networks were shown to perform extremely well, with quite low false alarm rates. This was quite a substantial achievement since these two competitions's datasets were designed to be as diverse as possible regarding their camera, ISO and processing pipelines. A possible conclusion after these events was that the problem of cover-source mismatch was not really relevant anymore since deep neural network were able to cope with diversity. This paper proposes to study

The work presented in this paper received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101021687 (project "UNCOVER").

the actual truthfulness of this claim.

Our series of work [13], [14] was the first systematic study of the impact of the different properties of natural images on cover-source mismatch. This allowed a clearer understanding of what properties of natural images actually mattered in steganalysis. The conclusion was that cover source mismatch rested on three main properties of images: the camera sensor, the ISO setting and the processing pipeline.

This work also recalled two main strategies to mitigate the impact of cover source mismatch:

- **Atomistic approach**: This approach is based on sub-dividing a large dataset into smaller subsets containing images with similar properties. The idea is then to train specialized detectors for each of this smaller datasets. This approach is inspired by forensic-aided steganalysis methods such as [15], [16] where the authors used forensic tools to build datasets with images having similar statistical properties.
- **Holistic approach**: This approach is based on making the training set as diverse as possible so that the rule learned by the classifier is the least dependent on image sources as possible. This strategy was explicitly proposed in [17] and applied in [18] and [19].

In [14], it was shown that the holistic approach was very costly sample-wise compared to the atomistic approach which led to the idea of training set design to improve on steganalysis.

The experiments of this work were however performed during the years 2017-2018, at a time where deep neural network were only starting to make their way into steganalysis. Consequently, we only used the older techniques based on rich models and classifiers. In this paper, we propose to update the results of these older works by using the state of the art approach of using EfficientNet with transfer learning.

II. Common experimental settings

Over the course of this paper, several datasets are built by fixing different parameters. Each dataset is always composed of 10,000 grayscale cropped JPEG cover images of dimension 264×264 and their 10,000 stego counterparts. A training set is built out of 70% of a given dataset, while the rest corresponds to the testing set. All images come from the ALASKA dataset [11], [12].

For each experiment, we also always produce a dataset composed of 10,000 images called MIX which correspond to the Holistic approach. It is built by taking the same number of images in each individual datasets so that the class of images generated from a given set of parameter is balanced with each other class.

Steganalysis is always performed by using EfficientNet-b3 [8] in its original configuration with the exception that the stem stride is set to 1. The starting learning rate is set to 0.25 and divided by 2 on each loss plateau. Due to some datasets being more difficult than others, we performed curriculum learning by starting on images embedded with a payload of 0.7 bits per DCT coefficients (bpc) followed by 0.5 bpc and finally 0.3 bpc. With the exception of an especially difficult processing pipeline, this ensured the convergence of the networks for each datasets.

Steganalysis performance is presented as a table of P_E – see Eq (1) :

$$P_E = \min_{P_{FA}} \frac{1}{2} \left(P_{FA} + P_{MD} \right), \tag{1}$$

where P_{FA} and P_{MD} are respectively the false alarm rate and the rate of missed detection of the detector on the given testing set. The P_E is obtained by computing a ROC curve from the soft outputs of the classifier. The rows of the table correspond to the training set and the columns to the testing set. The intrinsic difficulty of each dataset can be read on the diagonal of the table while the impact of cover-source mismatch – which we call **source inconsistency** in this paper – is read by columns. To facilitate the reading of the numerous results, we subtracted the P_E of non-diagonal entry by the P_E of its corresponding diagonal entry: each non-diagonal entry thus directly refers to the source inconsistency between the training and testing set.

III. Impacts of Camera and ISO setting

To study the impact of the camera we selected five cameras of varying quality which are presented in Table I. With the exception of the Canon EOS 500D, we have tried to keep the ISO relatively low, however one should understand that even if two cameras use the same ISO setting, the resulting noise will most certainly be different. As such, it is difficult to study the impact of the camera in isolation from the ISO. Therefore, the source inconsistency should here be interpreted as stemming from both the camera and ISO. All images were developed using Rawtherapee 5.8, in its default settings using the Amaze demosaicking algorithm. We chose to use the current state of the art steganographic algorithm for non side-informed JPEG steganography, J-UNIWARD [20]. Results are presented in Figure 1.

A first overall observation of these results is the large diversity of both intrinsic difficulty and source inconsistency even when the processing pipeline is fixed. It is difficult to attribute the intrinsic difficulty to either the camera or the ISO setting. For example, the iPad Pro and the HTC One A9, being handheld devices, have both low quality camera sensors and yet, despite the ISO of the HTC being higher than the ISO setting of the iPad, they both have the same intrinsic difficulty. On the other hand, the Canon EOS 500D, which has quite a

TABLE I
CHARACTERISTICS OF THE DIFFERENT CAMERA SENSORS USED IN THE EXPERIMENTS OF THIS CHAPTER.

Camera name	ISO	Year	Sensor size (mm)	Megapixels
Canon EOS 500D	1600	2009	22.3×14.9	15.1
Lumix DMC-GM1	200	2013	17.3×13.0	16
HTC One A9	93	2015	7.1 (total)	13
Apple iPad Pro	20	2015	4.80×3.60	12
Nikon D610	100	2013	35.9×24	24.3

high ISO setting compared to the other cameras, clearly has the highest intrinsic difficulty among these datasets. Consequently, we can conclude that both parameters play an important role here.

Regarding source inconsistency, it is almost always larger than 5% irrespective of the camera. An interesting case is that of the Nikon D610, the highest quality camera among those studied here. First it has the lowest intrinsic difficulty compared to the other cameras. This is most likely due to the quality of the sensor leading to almost noiseless images at low ISO. Secondly, it has a somewhat low source inconsistency with other sources but it leads to very high inconsistency with other datasets when it is used as the training set. This shows that even if a dataset has low source inconsistency with other sources, it might still be an extremely bad choice as a training set. Finally, the MIX strategy seems excellent at mitigating the impact of CSM in this case as it always leads to smallest source inconsistency when used as a training set.

IV. ISO SENSITIVITY

The impact of the ISO is easier to study in isolation by fixing the camera. Therefore we used two in-house datasets termed M9Base1 and M9Base2 taken with a single Leica M9 camera. These dataset were made by photographing *exactly the same scenes* at different ISO which allows us to isolate the impact of the ISO from every other parameter. Note that scenes differ between M9Base1 and M9Base2 datasets. Once again, all images were developed using Rawtherapee 5.8, in its default settings. Results are presented in Figure 2-3.

As expected, the higher the ISO, the higher the intrinsic difficulty of the dataset. See for example in Table 1 where we go from an intrinsic difficulty of 10.5% at ISO160 up to 18.7% even though both the camera and the content of images are fixed. Also notice that semantic content does play a role here as images in M9Base2 taken at ISO1000 have an intrinsic difficulty similar to M9Base1 taken at ISO640.

In this setting, note that the source inconsistency is always the lowest when using the MIX strategy, that is when training on a dataset where all the different ISO are present. However it should be noted that even in this case, source inconsistency can still be pretty high with values which can go up to 5% in the case of ISO640.

V. PROCESSING PIPELINE

Finally, in order to study the impact of the processing pipeline, we fixed the camera and ISO of each dataset while performing different kinds of processing operations for each dataset.

Seven different processing pipelines using either Rawtherapee 5.8 (RT) or the *rawpy* library were chosen. In the case of Rawtherapee, each pipeline uses the default settings (using the Amaze demosaicking algorithm) while varying a single algorithm. In the case of *rawpy*, every setting is turned off, except for the white balance which is set in camera mode.

We now describe each of the pipelines:

- Three demosaicking algorithms: Amaze (RT), Bilinear (rawpy) and PPG (rawpy). The Amaze algorithm is one of the current state of the art among (open-source) demosaicking algorithms. The PPG algorithm is a simpler and faster algorithm which is used to generate BossBase. Finally, the bilinear algorithm (simplified as LIN) is the simplest and fastest non-trivial demosaicking algorithm possible, at the cost of large loss of quality compared to the other two.
- One sharpening algorithm from RT with two sets of parameters – USM soft and USM hard. The algorithm is a modified version of the classic Unsharp Mask algorithm [21] used to enhance the edges and contrast of an image. The first set only applies soft sharpening while the second applies very aggressive edge enhancement. The latter amplified the noise so much that our network usually did not converge due to resulting difficulty of the dataset. In theses cases, we omit to present the results.
- One denoising algorithm from RT with two sets of parameters – DEN soft and DEN hard. The algorithm uses the Directional Pyramid Denoising based on wavelet decomposition [22]

	MIX	Canon 500D	DMC-GM1	HTC A9	Ipad	Nikon D610
MIX	10.6	+3.6	+2.2	+2.9	+3.3	+0.9
Canon 500D	+12.7	16.8	+27.9	+18.9	+18.1	+13.0
DMC-GM1	+6.4	+15.2	6.0	+9.4	+5.1	+1.1
HTC A9	+8.4	+5.5	+10.1	10.5	+7.4	+9.0
Ipad	+9.2	+5.4	+4.5	+7.8	10.2	+3.5
Nikon D610	+19.8	+27.5	+16.1	+26.4	+23.7	1.6

Fig. 1. Table of P_E for different cameras embedded with J-UNIWARD at payload 0.3bpc.

	MIX	ISO160	ISO320	ISO640
MIX	19.3	+2.3	+1.5	+5.0
ISO160	+1.8	10.5	+2.0	+8.9
ISO320	+2.9	+4.5	7.3	+8.0
ISO640	+0.5	+9.7	+4.5	18.7

Fig. 2. Table of P_E for different ISO on M9Base1 embedded with J-UNIWARD at payload 0.3bpc.

	MIX	ISO500	ISO1000	ISO1250
MIX	17.5	+3.2	−0.9	+1.8
ISO500	+2.0	15.5	+2.1	+5.6
ISO1000	+3.8	+7.9	18.0	+2.0
ISO1250	+2.0	+8.1	−1.8	19.9

Fig. 3. Table of P_E for different ISO on M9Base2 embedded with J-UNIWARD at payload 0.3bpc.

	MIX	Amaze	LIN	PPG	USM soft	USM hard	DEN soft	DEN hard
MIX	10.0	+3.4	+1.0	+1.3	+1.5	*	+1.9	+0.6
Amaze	+14.4	10.2	+14.4	+24.2	+3.5	*	+8.8	+28.0
LIN	+31.6	+32.4	0.3	+37.3	+26.1	*	+34.4	+34.2
PPG	+19.6	+30.0	+15.8	2.7	+23.7	*	+30.5	+22.2
USM soft	+18.8	+6.7	+14.8	+34.6	19.5	*	+18.9	+33.0
USM hard	+39.8	+39.6	+49.0	+46.8	+30.3	*	+46.9	+48.8
DEN soft	+15.0	+15.5	+11.9	+30.7	+14.7	*	3.0	+17.4
DEN hard	+19.3	+33.8	+13.4	+34.5	+26.4	*	+10.0	0.6

Fig. 4. iPad Pro – ISO 20. Table of P_E for different processing pipelines embedded with J-UNIWARD at payload 0.3bpc and steganalysis performed with EfficientNet-b3. A column is starred (*) if EfficientNet did not converge for the cell on the diagonal.

	MIX	Amaze	LIN	PPG	USM soft	USM hard	DEN soft	DEN hard
MIX	17.1	+6.0	+0.0	+3.8	+0.4	*	+4.5	+0.3
Amaze	+17.4	16.8	+34.1	+27.0	+7.8	*	+17.4	+13.1
LIN	+23.0	+33.2	0.1	+48.0	+13.9	*	+45.2	+34.2
PPG	+20.8	+30.9	+30.0	1.5	+13.8	*	+40.1	+38.1
USM soft	+29.5	+18.3	+47.3	+46.3	36.0	*	+45.9	+47.9
USM hard	+32.5	+33.0	+49.6	+47.6	+13.8	*	+47.4	+49.7
DEN soft	+17.9	+18.7	+13.7	+28.9	+8.7	*	2.3	+4.2
DEN hard	+17.0	+33.1	+5.6	+48.5	+13.9	*	+30.6	0.1

Fig. 5. Canon EOS 500D – ISO 1600

A first overall observation is that the impact of the processing pipeline on Cover-Source Mismatch is dramatic, with source inconsistency which can reach 49% even though the camera, the ISO and the scenes present in the datasets are identical. This observation was already made in [14] but it was hoped that using neural network would allow for better generalizations between sources which is clearly not the case in practice. A good news however is that using a training set which includes all the possible processing pipelines does allow for better generalization, even though we kept the number of samples identical for all datasets. This generalization results should however be studied more thoroughly as the diversity of processing pipeline "in the wild" can get quite difficult for the steganalyst to handle.

Now, going to an analysis of each individual processing pipelines, we can observe that pipelines which amplifies details and edges – USM and the Amaze algorithm – lead to higher intrinsic difficulties than pipeline which tend to smooth the image such as denoising and linear demosaicking. This is to be expected, the more textured an image is, the more difficult it is to model its content and thus to separate it from the stego signal.

It is also interesting to note that processing pipelines which perform similar operations but with different parameters tend to have lower source inconsistency. For example, in the case of the Nikon D610 camera, the two USM pipelines have a source inconsistency no higher than 1.5% but higher than at least 4% for every other pipeline except Amaze. All results are shown for different cameras in Figure 4-8.

	MIX	Amaze	LIN	PPG	USM soft	USM hard	DEN soft	DEN hard
MIX	9.3	+3.7	+2.4	+2.7	+3.7	+3.2	+3.1	+1.6
Amaze	+8.3	6.0	+18.4	+23.9	+2.7	+4.2	+6.2	+20.2
LIN	+24.8	+29.5	0.5	+29.7	+30.5	+23.7	+28.5	+19.1
PPG	+15.1	+24.3	+19.4	3.3	+25.2	+19.1	+16.3	+9.3
USM soft	+8.5	+2.6	+10.5	+27.9	11.0	+2.0	+8.2	+18.9
USM hard	+11.0	+3.6	+18.6	+31.3	+1.2	19.4	+9.1	+33.1
DEN soft	+7.9	+11.2	+8.5	+18.2	+13.3	+10.1	2.7	+3.1
DEN hard	+12.9	+26.9	+12.6	+13.4	+23.5	+18.7	+7.6	1.0

Fig. 6. DMC-GM1 – ISO 200

	MIX	Amaze	LIN	PPG	USM soft	USM hard	DEN soft	DEN hard
MIX	13.6	+3.4	+1.3	+2.4	+4.1	*	+1.8	+1.1
Amaze	+13.7	10.5	+36.5	+25.5	+9.2	*	+7.9	+21.0
LIN	+28.5	+31.2	0.1	+39.2	+25.0	*	+38.3	+31.5
PPG	+22.4	+33.3	+34.9	3.1	+25.3	*	+33.3	+28.0
USM soft	+10.9	+3.7	+11.2	+26.5	22.4	*	+9.9	+13.1
USM hard	+35.9	+39.3	+48.7	+45.8	+27.4	*	+46.6	+49.3
DEN soft	+18.8	+22.1	+30.0	+36.1	+19.7	*	3.0	+12.3
DEN hard	+17.4	+34.8	+16.0	+34.3	+25.4	*	+10.6	0.6

Fig. 7. HTC One A9 – ISO 93

VI. Impact of the Image Content

The main thesis resulting from of our original cover-source mismatch study in [14] is that the source of cover-source mismatch is the differences of noise properties of images between the training and testing sets. This lead us to focus on the camera sensor, ISO setting and processing pipelines.

However, another element of natural images which might play a role in cover-source mismatch is the actual content of an image. Intuitively, if a training dataset consists only of very smooth images such as blue skies whereas the testing set is made out of very textured images, one might expect significant cover-source mismatch. To test this hypothesis we performed exactly this experiment by creating four datasets each made out of 10,000 images. Each dataset is made out of images captured with a Sony Alpha 7 camera and developed either with the USM soft processing pipeline or the DEN soft processing pipeline as described in Section V. Furthermore each dataset contains only images selected to be smooth (SMO) or textured (TEX).

The steganography and steganalysis is performed in the same way as for the previous experiments. Results appear in Figure 9.

Looking at these results it is important to highlight that even if datasets share a processing pipeline and a camera, we can still observe the impact of cover-source mismatch. In this experimental setting, which is a pretty extreme case of "content mismatch", we observe source inconsistency of up to 13.6% for the USM soft pipeline and 12.5% for the DEN soft pipeline.

On the other hand, it is also important to note that inconsistencies when the pipeline is fixed are always **the lowest** inconsistencies we observe in the experiment. For example, the source inconsistency between DEN and USM reaches 31.6%

	MIX	Amaze	LIN	PPG	USM soft	USM hard	DEN soft	DEN hard
MIX	3.3	+2.1	+0.5	+0.6	+2.8	+2.3	+0.8	+0.8
Amaze	+1.7	1.6	+2.0	+4.1	+1.0	+1.4	+7.6	+1.1
LIN	+11.8	+20.0	0.2	+12.8	+20.9	+19.5	+5.7	+3.0
PPG	+2.5	+6.4	+0.9	0.5	+8.0	+8.0	+0.7	+0.9
USM soft	+0.7	+0.8	+1.5	+3.9	2.8	+1.4	+4.6	+3.4
USM hard	+1.6	+1.0	+2.5	+2.8	+0.8	5.7	+4.5	+3.5
DEN soft	+0.5	+2.5	+0.9	+1.7	+3.9	+3.7	1.4	+0.3
DEN hard	+3.6	+7.2	+2.6	+8.4	+6.2	+5.0	+1.7	0.6

Fig. 8. NIKON D610 – ISO 100

	USM soft | TEX	DEN soft | TEX	USM soft | SMO	DEN soft | SMO
USM soft | TEX	29.3	+31.6	+6.2	+12.1
DEN soft | TEX	+13.4	3.4	+25.3	+1.3
USM soft | SMO	+11.6	+16.0	7.0	+0.8
DEN soft | SMO	+19.7	+12.5	+11.2	0.5

Fig. 9. Sony alpha 7. Table of P_E for different processing pipelines and different content types for images embedded with J-UNIWARD at 0.3bpc. The SMO datasets only contain smooth images while the TEX dataset contain texture images.

for the TEX while it only of 12.5% if we train on DEN TEX and test on DEN SMO.

The conclusion is similar to what was found in [14]: content can lead to cover-source mismatch but differences in the processing pipeline largely dominate.

VII. Conclusion and perspectives

This paper is a short update on the systematic study on cover-source mismatch using the current state of the art in steganalysis. The results of this paper paint a more nuanced picture. Indeed a common observation on all four properties of natural image studied herein – camera, ISO, processing pipeline and content – is that cover-source mismatch is always very strong with EfficientNet, provided that the testing set contains images with different properties than the training set. On the other hand, the holistic/MIX strategy, that is using a training dataset containing a diverse set of properties, is very effective in mitigating the effect of cover-source mismatch even with datasets which are quite small.

This a real difference with steganalysis using rich models where we observed in [14, Sec. 9] that the holistic strategy demanded a lot more samples to perform as well as the case where no mismatch is present. Obviously, the experiments presented in this paper are somewhat limited to a single (though prevalent) steganographic algorithm – J-UNIWARD – and to a single deep network architecture – EfficientNet-b3. However, we believe this setup is sufficiently representative of the current environment of steganography and steganalysis so that these results should generalize to other modern algorithms.

References

[1] T. Pevny, P. Bas, and J. Fridrich, “Steganalysis by subtractive pixel adjacency matrix,” *IEEE Transactions on information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[2] P. Bas, T. Filler, and T. Pevný, “Break Our Steganographic System — the ins and outs of organizing BOSS,” in *Information Hiding, 13th International Workshop*. Prague, Czech Republic: vol.6958, Springer, May 2011, pp. 59–70.

[3] V. Holub and J. Fridrich, “Low-Complexity Features for JPEG Steganalysis Using Undecimated DCT,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, Feb. 2015.

[4] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, “Steganalysis of Adaptive JPEG Steganography Using 2D Gabor Filters,” in *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*. Portland Oregon USA: ACM, Jun. 2015, pp. 15–23.

[5] G. Xu, H.-Z. Wu, and Y.-Q. Shi, “Structural Design of Convolutional Neural Networks for Steganalysis,” *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, May 2016.

[6] G. Xu, “Deep Convolutional Neural Network to Detect J-UNIWARD,” in *IH&MMSEC 5*. Philadelphia, USA: ACM, Jun. 2017, pp. 67–73.

[7] M. Boroumand, M. Chen, and J. Fridrich, “Deep Residual Network for Steganalysis of Digital Images,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, May 2019.

[8] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *arXiv:1905.11946 [cs, stat]*, Sep. 2020, arXiv: 1905.11946.

[9] M. Goljan, J. Fridrich, and T. Holotyak, “New blind steganalysis and its implications,” E. J. Delp III and P. W. Wong, Eds., San Jose, CA, Feb. 2006, p. 607201.

[10] J. Kodovský, V. Sedighi, and J. Fridrich, “Study of cover source mismatch in steganalysis and ways to mitigate its impact,” A. M. Alattar, N. D. Memon, and C. D. Heitzenrater, Eds., San Francisco, California, USA, Feb. 2014, p. 90280J.

[11] R. Cogranne, Q. Giboulot, and P. Bas, “The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis,” in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*. Paris France: ACM, Jul. 2019, pp. 125–137.

[12] ——, “ALASKA#2: Challenging Academic Research on Steganalysis with Realistic Images,” in *2020 IEEE WIFS*. New York, USA: IEEE, Dec. 2020, pp. 1–5.

[13] Q. Giboulot, R. Cogranne, and P. Bas, “Steganalysis into the Wild: How to Define a Source?” *Electronic Imaging*, vol. 2018, no. 7, pp. 318–1–318–12, Jan. 2018.

[14] Q. Giboulot, R. Cogranne, D. Borghys, and P. Bas, “Effects and solutions of Cover-Source Mismatch in image steganalysis,” *Signal Processing: Image Communication*, vol. 86, p. 115888, Aug. 2020.

[15] M. Barni, G. Cancelli, and A. Esposito, “Forensics aided steganalysis of heterogeneous images,” in *2010 IEEE ICASSP*, Mar. 2010, pp. 1690–1693, iSSN: 2379-190X.

[16] X. Hou, T. Zhang, G. Xiong, and B. Wan, “Forensics aided steganalysis of heterogeneous bitmap images with different compression history,” in *Conference on Multimedia Information Networking and Security*. IEEE, 2012, pp. 874–877.

[17] I. Lubenko and A. D. Ker, “Steganalysis with Mismatched Covers: Do Simple Classifiers Help?” in *Proceedings on Multimedia and Security*, ser. MM&Sec ’12. New York, USA: ACM, 2012, pp. 11–18.

[18] A. D. Ker and T. Pevný, “A mishmash of methods for mitigating the model mismatch mess,” in *Media Watermarking, Security, and Forensics 2014, Proc. SPIE*, vol. 9028, 2014, pp. 1601–1615.

[19] J. Pasquet, S. Bringay, and M. Chaumont, “Steganalysis with cover-source mismatch and a small learning database,” in *2014 22nd European Signal Processing Conference (EUSIPCO)*. IEEE, 2014, pp. 2425–2429.

[20] V. Sedighi, R. Cogranne, and J. Fridrich, “Content-Adaptive Steganography by Minimizing Statistical Detectability,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, Feb. 2016.

[21] A. Polesel, G. Ramponi, and V. Mathews, “Image enhancement via adaptive unsharp masking,” *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 505–510, Mar. 2000.

[22] T. Nguyen and S. Oraintara, “The Shiftable Complex Directional Pyramid—Part II: Implementation and Applications,” *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4661–4672, Oct. 2008.