https://doi.org/10.2352/ISSN.2470-1173.2018.07.MWSF-318

Steganalysis into the Wild: How to Define a Source?

***Quentin Giboulot*°, *Rémi Cogranne*° *and Patrick Bas**.**
° ***Lab. of System Modelling and Dependability, ROSAS Dept., ICD, UMR 6281 CNRS, Troyes University of Technology, France.***
* ***CNRS, École Centrale de Lille, University of Lille, CRIStAL Lab. , France.***

Abstract

It is now well known that practical steganalysis using machine learning techniques can be strongly biased by the problem of Cover Source Mismatch. Such a phenomenon usually occurs in machine learning when the training and the testing sets are drawn from different sources, i.e. when they do not share the same statistical properties. In the field of steganalysis however, due to the small power of the signal targeted by steganalysis methods, it can drastically lower their performance.

This paper aims to define through practical experiments what is a source in steganalysis. By assuming that two cover datasets coming from a common source should provide comparable performances in steganalysis, it is shown that the definition of a source is more related with the processing pipeline of the RAW images than with the sensor or the acquisition setup of the pictures. In order to measure the discrepancy between sources, this paper introduces the concept of consistency *between sources, that quantifies how much two sources are subject to Cover Source Mismatch. We show that by adopting "training design", we can increase the consistency between the training set and the testing set. To measure how much image processing operation may help the steganographers this paper also introduces the* intrinsic difficulty *of a source.*

It is observed that some processes such as JPEG quantization tables or the development pipeline can dramatically increase or decrease the performance of steganalysis methods and that other parameters such as the ISO sensitivity or the sensor model have minor impact on the performance.

Introduction

The security of steganography algorithms as well as the benchmark of steganalysis schemes is usually evaluated on the well-known BOSSBase [1] generated following a unique processing pipeline. This setting has indisputable advantages for both steganography – allowing the comparison between steganographic schemes, choosing parameters of embedding scheme to maximize efficiency [16] – and steganalysis – designing of features sets [6, 17] and studying the impact of several parameters on detectability [21]. However, this methodology that uses the same dataset, with limited diversity, and that processes all raw images using exactly the same processing pipeline has several important limitations. Indeed, such an experimental framework seems quite far from a real-life environment where images come from many different camera models with a possible wide range of acquisition setups (e.g. different sensors, different ISO sensitivity or ISO speed, ...) and are subject to different processing pipelines.

This paper shows how these discrepancies impact on the phenomenon of cover-source mismatch (CSM) which can be loosely defined by the fact that if the training and testing sets do not come from the same source, steganalysis then undergoes a strong loss in accuracy.

To the best of our knowledge, very few works (see for instances [2, 14, 11, 12]) have tried to characterize the sources of the CSM, quantify their impact and address it. Note that those works mostly focus on the image acquisition settings such as the camera model and the ISO sensitivity. A notable prior work, however, is [21] in which the authors show that for spatial domain image steganography, cropping or resizing significantly changes the performance of steganalysis methods. Although CSM has only been studied in a handful of prior works, this problem is fundamental to address the larger problem of real-life scenarios, as already acknowledged in [10], and will be beneficial both for the steganalyst and the steganographer. The former must understand which acquisition or processing parameters have the largest impacts on classification accuracy, the latter must understand those parameters to choose images for which the hidden message will be harder to detect.

Contents of the Paper

We recall first the outline of the paper:

- Section “Experimental Setup” defines the classification setup and the studied parameter that can possibly impact the mismatch.
- Section “Steganalysis on Real-Life Image Bases” motivates our study by presenting steganalysis results on databases coming from different sensors, at different resolutions, or at different ISO sensitivity.
- Section “Training Design” studies the impact of different parameters such as the JPEG Quality Factor, the camera sensor, the processing software, processing algorithms, ISO Sensitivity or possible color adjustments.
- Section “Co-occurrence Analysis of Different Development Settings” attempts to give a statistical rational on the problem of CSM by looking at the cooccurence of neighboring pixels after distinct development pipelines.

- Finally section “Conclusion” lists the parameters that have either minor or major impacts on the mismatch.

Experimental Setup

Throughout this paper we follow a classic supervised classification setting, composed of training and testing sets where each cover image is paired with its stego-image. Both training and testing set are composed of 5000 random images from their corresponding image base. More specifically we use the low complexity linear classifier defined in [4] with five fold cross-validation to estimate the regularization parameter. We choose this classifier over the well-known ensemble [13] for its low computational costs which allows us to speed up the classification without – according to the results given in [4, 3] – loosing in terms of steganalysis accuracy. Experiments are conducted on images compressed using the JPEG standard and two well-known embedding schemes have been used, namely NSF5 [7] a rather old non-adaptive steganographic algorithm, and the content-adaptive scheme J-UNIWARD [9]. For feature extraction, the DCTR [8] algorithm has been used.

Measure of Cover-Source Mismatch

To be able to quantify the impact of Cover-Source Mismatch (CSM), one first needs a definition of a source. We will define a priori a source as two sets of parameters used to generate a natural image :

Acquisition parameters This encompasses all parameters fixed during the acquisition of the raw image by a camera, e.g. camera model, ISO, sensor size, etc. ...

Processing parameters This encompasses the whole processing pipeline after the image has been taken, e.g. demosaicing algorithms, resampling, cropping, processing software, JPEG compression, etc. ...

We will refine those definitions in Section "Training Design" to the sole parameters which have an impact on steganalysis accuracy.

Once a source has been defined, two important properties related to the image bases must be introduced:

- The probability of error given that the training and testing sets both come from the same source, this is defined as the *intrinsic difficulty* of the image base. The steganographer will for example seek for sources with the highest intrinsic difficulty.
- The probability of error given that the training and testing sets each comes from a different source, this is defined as the *inconsistency*, or source mismatch, between training and testing sets. A high inconsistency inducing an important mismatch between the two databases, the steganalyst will consequently try to generate a training database providing a low inconsistency with the given testing image.

Dataset #	Camera Model	Fixed Dimension	ISO
1	Nikon D90	—	—
2	Nikon D90	—	200
3	Nikon D90	4288×2848	200
4	—	5184×3456	—
5	—	—	500
6	—	—	1600

Table 1. Summary of selected subset of the FlickRBase. Every database also had a Quality Factor (QF) fixed to 99

Note that the probability of error is measured using the most used minimal total probability of error under equal priors $P_E = \min(P_{FA} + P_{MD})/2$.

Steganalysis on Real-Life Image Bases

To understand the necessity of a finer characterization of the CSM phenomenon, one must first confront classical steganalysis techniques to real life databases, that is image bases which have the following properties :

1. They contain images with numerous different acquisition parameters (sensors, ISO, exposition, etc. ...).
2. Each image has potentially followed a specific processing pipeline (specific compression parameters, processing steps, image editing software, etc. ...).
3. The processing history is a priori unknown.

To that end, we used the FlickRBase [20] which contains 1.3 millions images downloaded from FlickR in their original quality (this ensures that no further compression was applied after uploading, which would normalize the image base). Acquisition parameters were associated to each image using their EXIF data[1].

From this image base, we constructed several databases consisting of 10 000 images with one or more fixed acquisition parameters, they are summarized in Table 1. Images were then losslessly center-cropped to get images of size 512×512 using the command *jpegtran*[2], to ensure that the 8×8 block structure of jpeg files is preserved. Each base was then classified with the methodology exposed in the previous Section, images being embedded using NSF5 with payload 0.1 bpnzac. Some results obtained with a few fixed parameters are presented in Table 2. Note that, for readability, only a handful of results are presented here; very similar results were obtained with much more datasets with various sets of fixed parameters (camera model, ISO sensitivity, shutter speed, sensor size, aperture, etc. ...).

We can immediately observe that the intrinsic difficulty is far from the BOSSbase baseline[3] despite fixing several acquisition parameters usually associated to the causes

[1] Yahoo Flickr Creative Commons 100M (YFCC100M) is available freely at https://webscope.sandbox.yahoo.com; note that this dataset is hosted on the Amazon Web Services platform, which requires a free Amazon Web Services login for access.

[2] The command-line program *jpegtran* is part of *libjpeg* library which can be found at http://ijg.org.

[3] See [8, Fig.6] which claims P_E of about 20% for a Quality

of cover-source mismatch phenomenon (sensor, ISO, original image dimensions). This consequently implies that acquisition parameters are not sufficient to define a source. The fact that images coming from such platforms as FlickR are always heavily processed can, however, lead us to the idea that it is the diversity of processing pipelines that is the main culprit behind the high intrinsic difficulty of such databases. Indeed if each image composing an image base followed a different processing pipeline, then image properties when split between training and testing base will tend to be quite different.

To explore this idea and motivate the next section, we repeat the previous experiment except that instead of cropping the images, we downsample them to a size of 512×512 using the Lanczos filter of *convert*. Intuitively, since we take images in their original sizes, resizing them to 512×512 will have a huge impact on the image properties and on pixel distributions such that past processing will be negligible compared to the downsampling. This way, we normalize the processing parameters of each image base. The results in terms of steganalysis accuracy with the same datasets as those used presented in Table 3 using images resizing are summarized in Table 3.

Train\Test	1	2	3	4	5	6
1	**0.34**	0.34	0.34	0.39	0.35	0.35
2	0.34	**0.33**	0.34	0.37	0.34	0.34
3	0.35	0.35	**0.31**	0.36	0.34	0.35
4	0.39	0.37	0.40	**0.34**	0.35	0.36
5	0.34	0.34	0.35	0.34	**0.32**	0.33
6	0.35	0.35	0.37	0.34	0.33	**0.33**

Table 2. Intrinsic difficulty and consistence of cropped 512×512 image bases for different acquisition parameters. We show that classification accuracy is not significantly increased by simply fixing acquisition parameters for real-life image databases.

Train\Test	1	2	3	4	5	6
1	**0.14**	0.14	0.14	0.14	0.14	0.15
2	0.15	**0.13**	0.15	0.13	0.14	0.14
3	0.13	0.14	**0.15**	0.14	0.14	0.14
4	0.14	0.13	0.15	**0.11**	0.13	0.14
5	0.14	015	0.14	0.14	**0.13**	0.15
6	0.15	0.14	0.14	0.14	0.15	**0.14**

Table 3. CSM matrix of downsampled 512×512 images with a fixed acquisition parameter downloaded from Flickr.

From the significant drop in P_E from Table 2 Table 3 we can infer that the processing pipeline has a huge impact.

This section showed that current good practices are far from sufficient to allow accurate steganalysis in a supervised setting. Fixing acquisition parameters such as the sensor, ISO and dimension of the image does not give a satisfying jump in accuracy. However, it is clear from the difference in intrinsic difficulty between cropped and resized images, that the way an image is processed has a far bigger impact on the way the stego noise will be distributed in the image. Thus it is not sufficient to define a source as the acquisition parameters ; processing parameters must also be taken into account in this definition.

The following sections will try to demonstrate that not only these processing parameters have an impact, but also that the impact of the acquisition parameters are negligible compared to those of the processing parameters.

Training Design

A supervised classification problem always relies on at least three assumptions regarding the relationship between the training and testing base :

1. The marginal distributions of features extracted from the training and testing sets are close.
2. The conditional distributions of those features given the distribution of each class (here cover and stego) of the training and testing sets are close.
3. The marginal distributions of each class of the training and testing sets are close.

In a real life setting of steganalysis with highly diverse content, image processing tools and camera models, none of those assumptions are accurate. To be able to use the classical supervised classification setting in such a steganalysis context one must design a methodology to ensure that, for a set of inspected digital images, an ad-hoc training set is able to verify these conditions. In this work, we will only deal with the first two assumptions which are directly related to the problem of cover-source mismatch. The proposed methodology is based on the fact that the steganalyst can have access to the RAW images that generates the training set. Thus, the steganalyst's problem is then to do **training design**, i.e. to find a methodology for designing a suitable subset of the training set such as to maximize accuracy of the classification of the testing set given a knowledge of the images properties (sensor, ISO, JPEG QF, etc...) and given partial or total knowledge about the processing pipeline history.

In other words, the idea is to split the training base into training sub-bases coming from a fixed source and to associate to each sub-base a training base which would best approximate our two assumptions of interest.

For this method to work, one first needs a definition of a source. A good definition of a source would specify those parameters which are the biggest source of inconsistency (as defined in the Introduction) between training and testing base. The intrinsic difficulty would then define a lower bound on the accuracy given a specific setting (stego-algorithm, classifier, features).

The main hypothesis of this work is that the most effective steganalysis in our setting is achieved by training the classifier on training set which has followed the same processing pipeline as the testing set. Rephrasing using the aforementioned defined assumption on which supervised learning relies, we hypothesize that a source is well approximated by a specific processing pipeline and that the

Factor (QF) of 75 and slightly below 15% for QF 95 on BOSSbase using DCTR features set.

	Radius	Amount		Radius	Amount	Damping	Itérations		Luminance
USM1	1	300	RL1	0.75	75	20	30	DEN1	30
USM2	2.5	300	RL2	1.5	100	0	50	DEN2	40
USM3	0.5	550	RL3	2.5	100	0	50	DEN3	55
USM4	0.5	800	RL4	2.5	100	0	70	DEN4	70
USM5	3	1000						DEN5	90

Table 4. Parameters of the different RawTherappe 5.3 developpement settings studied. Parameters not specified were left to their default value.

loss in consistency incurred by not specifying the acquisition parameters is negligible.

To test for this hypothesis, we propose the following methodology :

1. A training set with RAW images taken with potentially several different sensors is selected.
2. A testing set with RAW images taken with a unique sensor not present in the training set is selected.
3. From both of these sets, $2N$ new sets are generated using N different processing pipelines in such a way that each new set follows a single processing pipeline and that each new training set follows the same processing pipeline as another testing set.

For our experiments, the training set will be the BOSSBase with images taken with an M9 Digital Camera removed and the testing set will be the M9Base which contains images taken only with a M9 Digital Camera at fixed ISO. After processing, every image is then converted to JPEG with a quality factor of 99 and, finally, losslessly center cropped with dimensions 512×512 using *jpegtran*. Note that we choose to fix the quality factor, or more precisely the quantization matrix, because this must be available in order to make the file usable. Therefore, since this parameter is known and, though it has an important impact on accuracy of steganalysis, it can hardly be a cause of source mismatch in practice. Similarly, the choice of cropping the images, instead of resizing, is justified from preliminary results on to FlickR base that studied the impact of various parameters on cover source mismatch. Indeed, eventually resizing the image by a important factor (typically to obtain images of size 512×512) largely reduces the impact of other processing.

The rest of this section will study the impact of two common acquisition parameters, namely the camera model and the ISO setting, coupled with the impact of processing parameters in an increasing level of detail. Two raw images datasets are used, the well-known BOSSbase which is made of 10 000 images from 7 camera models and an adhoc dataset that contain only images from a single camera model, namely the Leica M9, for which image has been captured fix few different values of ISO sensitivity parameter; this dataset is referred to as the M9Base.

Impact of Processing Softwares

The highest level of choice for processing parameters is the processing software. It will usually define the demosaicing algorithm, the white balance, the gamma correction as well as the JPEG compression parameters (chroma subsampling, quantification matrix, etc.). In addition an image editing software may use specific algorithms for the most common processing tools such as sharpening and denoising.

Table 5 provides results obtained using default settings with three different image editing softwares, namely *Photoshop Lightroom* (LR), a commercial high-end software for raw images editing and developed by *Adobe*, *RawTherapee* (RT) an open-source competitor of LR that allows both converting and processing raw images and *dcraw* (DC) an open source command-line program which provides only basic processing tools. Note that RawTherapee is based on *dcraw* for raw images conversion only.
A fourth dataset is included in Table 5 because LR uses custom JPEG quantization matrix (quality factor); we therefore choose to add a dataset referred to as LR-standardQF which is made of raw images converted to uncompressed TIFF format using LR and then compressed with standard QF using imagemagick *convert*'s tool[4].

Train \Test	LR-standardQF	LR	RT	DC
LR-standardQF	**0.234**	0.457	0.263	0.229
LR	0.459	**0.229**	0.474	0.450
RT	0.247	0.455	**0.251**	0.221
DC	0.284	0.450	0.301	**0.255**

Table 5. Intrinsic difficulty and consistence of images bases for different processing softwares. Each row corresponds to a training base while each column corresponds to a testing base. The training base is the BOSSbase with M9 images removed, the testing base is the M9Base-ISO1250

Train \Test	LR-standardQF	LR	RT	DC
LR-standardQF	**-0.004**	-0.001	0.002	0.005
LR	-0.003	**-0.003**	-0.001	0.002
RT	-0.004	-0.008	**-0.002**	-0.001
DC	-0.001	-0.000	-0.001	**0.005**

Table 6. Difference in P_E between training on the BOSSBase with M9 images removed and training on the same base as the testing set (here M9Base-ISO1250) a negative results means a better classification when training on the BOSSBase while a positive result means better classification when training on the same base as the testing set.

[4] The open-source software Imagemagick, including convert command-line tools, can be obtained at www.imagemagick.org.

Before presenting and commenting on the results, it is crucial to note that most steganography algorithms for JPEG compressed images embed a payload measured in bits per non-zero AC coefficients (bpnzac). However, the processing pipeline in general can heavily modifies the content details of a picture and, hence, the resulting number of non-zero AC coefficients. In order to conduct a fair comparison, we have thus chosen not to fix the embedding rate but instead to fix the length of the hidden data and to adjust the payload, in bpnzac, provided to the embedding scheme matches the message length. The payload is set to 10 322 bits or 1.26 Kilo-Bytes for nsF5, this corresponds to the payload of 0.04 bpac (Bits Per AC coefficients). We will report similar results in the final version of this paper for J-UNIWARD with embedded message length of 154 024 bits, or 15.75 Kilo-Bytes, corresponding to 0.6 bpac for J-UNIWARD. However, due to a much higher computation time for J-UNIWARD, the results reported corresponding to a payload of 0.5 bpnzac.

Results provided in Table 5 give two important lessons which can be summarized below:

- The most important parameter defining a source is first and foremost the quantification table used during the JPEG compression. Indeed when training on base with a different quantification than the testing base (e.g. LR/DC) we can see that the P_E always deviates by more than 20% from the intrinsic difficulty of the testing base.
- The inconsistency due to mismatched software alone is more subtle as it ranges from 0.4% (training on RT / testing on LR-standardQF) to roughly 5% (training on DC / testing on RT or LR-standardQF). However this amount is rather important as compared to the intrinsic difficulty which ranges from 23.4% (LR-standardQF) to 25.5% (DC).
- Each software has relatively close intrinsic difficulties, even with a non-standard quantification matrix. This means that the choice of software has only a negligible impact on the difficulty of an image when using all default parameters.
- The camera model does not seem to be a relevant parameter to characterize a source. Indeed, the absence of the M9 Camera from the training base does not prevent a rather low intrinsic difficulty for any of the image base. In order to confirm the negligible impact of the camera model on the definition of a source, Table 6 shows the difference in terms of steganalysis accuracy when trained on BOSSbase without the image from Leica M9 camera as compared to the results obtained when both training and testing are carried out on images from M9Base with the same ISO sensitivity. The loss of accuracy shows a negligible impact of both camera model and ISO sensitivity since the loss of accuracy with the same QF ranges from -0.8% to $+0.5\%$ for some cases in which the accuracy actually improves when trained on BOSSbase (when testing set has been edited with dcraw for instance).

Impact of Common Image Processing Algorithms

The comparison between image editing software only gives a coarse-grain view since, in all softwares, many processing algorithms can be tuned manually (for instance, the most important being the gamma correction, denoising, sharpening, and color adjustment). To study the impact of all those image processing algorithms individually we propose to choose one image editing software and to modify the parameters for the aforementioned various image processing algorithms. For each setting we then measure the *intrinsic difficulty* as well as the *consistency* between different datasets.

We choose RawTherapee (RT) as the image editing software because it is open source, which means that we exactly know the algorithms used and their parameters, and because it offers a wide range of parameters for each image processing algorithm. In the present paper we **only** present the results with the most influential and the common image processing tools, namely denoising, sharpening, and color adjustment. To measure the impact of the processing algorithms on both intrinsic difficulty and *inconsistency*, we present the results obtained by generating 15 different processing pipelines using:

- default settings, here we let the software automatically sets the processing pipeline, referred to as AutoLevels (AL);
- Unsharp mask (USM), which is an image sharpening method, with 5 different parameters for “radius” and “amount”;
- Richardson–Lucy (RL), a deconvolution image sharpening technique using a Gaussian Point-Spread Function (PSF), with 4 different parameters;
- Denoising (DEN), the noise reduction tool based on both wavelet decomposition and median filtering, with 5 different parameters.

The results in terms of steganalysis accuracy for all the image processing algorithms are summarized in Tables 10 – 15. For all those tables, each row corresponds to a specific dataset used for training and each column corresponds to the dataset used for testing. The training set is always obtained using RAW developed BOSSbase images, without the images from the camera model Leica M9, while the testing is always carried using RAW developed images from the M9Base.

First, Table 10 presents the results obtained when testing images with ISO sensitivity 160 using nsF5 embedding scheme. We summarize below the most important points that can be concluded from those results:

- It is always optimal (within 1%) to train and test on bases with the same development setting. That is, setting the same processing pipeline for the training and testing bases seems to be a sufficient (but by no means necessary) condition to get the best accuracy given a classification setting.
- Some development settings impact highly the intrinsic difficulty of the base; see for instance Unsharp mask, USM, for which P_E ranges from 31.5% to 39%, without source mismatch in Table 10 or Denoising (DEN)

for which intrinsic difficulty can be modified by a factor of about 10%.

- However the inconsistency between other datasets subject to the same type of processing is in general rather small.
- More generally, the inconsistency between all datasets heavily depends on the type of processing: this can be observed by observing P_E obtained when using each dataset for testing (on the column of Tables). While Unsharp mask (USM) and RL deconvolution seem both slightly sensitive to the training set – with standard deviation of P_E in the range $4 \sim 5\%$–, the denoising may be much more subjected to inconsistency and it seems very sensitive to the training set with standard deviation of P_E up to 15%.

Table 11 shows the difference in P_E when the training is carried out over images from the M9Base with ISO sensitivity 160, the dataset used for testing, as compared to the results obtained when training uses images from BOSSBase, as reported in Table 10. Once again we see that training on a base which contains the same camera model does not seem to help classification accuracy. Indeed the difference in P_E is, on average, -2%, meaning that it is on average more efficient to train on the BOSSBase with M9 image removed in this setting. While this might be due to semantic content differences, we can at least conclude that the camera model has a very weak impact on CSM compared to processing parameters. Note, however, that when there is no source mismatch in terms of processing pipeline, i.e. the diagonal of Tables 10 and 11, the intrinsic difficulty is slightly reduced by using the same raw images in almost all the cases. This, however, brings a rather small improvement for steganalysis as P_E is reduced in average by roughly 0.8% and at most by at most roughly 1.8%.

Next, Tables 12 and 13 present the very same results as those presented in Tables 10 and 11 only using J-UNIWARD with payload 0.5 bpnzac instead of nsF5 at payload 0.04 bpac. Though the embedding payload is much higher, the accuracy of steganalysis is greatly reduced since J-UNIWARD is the current state-of-the-art of adaptive embedding scheme, while on the opposite nsF5 is a non-adaptive scheme based on the F5 algorithm proposed in [18] in 2001. We note from Table 12 that the conclusion drawn from the results using nsF5 remains valid for J-UNIWARD especially on the impact of all the image processing algorithms on both intrinsic difficulty as well as on the inconsistency of datasets generated using different processing methods.
Similarly, we note from Table 13 that the conclusion from the comparison of results obtained by training on BOSSBase or on the same M9Base also remains valid ; the fact of picking the same camera model does not impact the inconsistency between image processing operations and only slightly (by at most 1.5%) the intrinsic difficulty.

Impact of ISO Sensitivity

Eventually, we propose to conduct the same experiment but modifying the ISO sensitivity of images used in the testing set from M9Base. The goal is obviously to measure how much the ISO sensitivity improves both intrinsic difficulty and inconsistency between different processing algorithms. Note that the ISO sensitivity corresponds, roughly speaking, to a signal gain; the higher the ISO sensitivity, the higher the noise standard deviation of pixels, and hence, of DCT coefficients.
To study the impact of ISO sensitivity, Tables 14 and 15 present the very same results as those presented in Tables 10 and 11 only changing the testing sets by images from the M9Base with ISO set to 1250, instead of 160. The comparison of those tables allows drawing the main following conclusions:

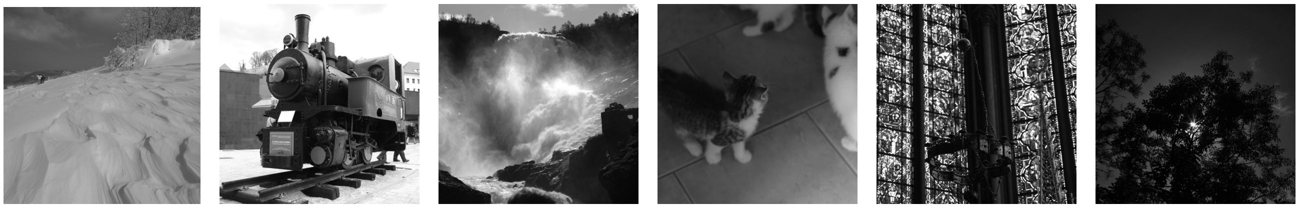
- First, and most important, moving from ISO 160 to ISO 1250 does not seems to have a strong impact neither on the intrinsic difficulty nor on the inconsistency between processing algorithms. Indeed the difference in terms of steganalysis P_E is around 0.5%.
- However this general results can vary according to the processing algorithms ; in general, when using Unsharp mask sharpening technique, the difficulty is increased with ISO of about $2 \sim 4\%$.
- The source mismatch, or inconsistency, between image processing settings is mostly not impacted by ISO sensitivity. Indeed, almost every inconsistency score differ by less than 1% between the testing on the M9Base-160 and on the M9Base-1250, the only exception being between DEN5 and RL development settings where the difference in inconsistency between ISO can reach 5%.
- The intrinsic difficulty is almost always slightly increased with the ISO; in average by 1.5% and up to 4.5% for the Unsharp mask algorithms. This seems rather natural since using a strong sharpening processing on noisy images tends to increase more the noise present in the image.

Impact of Color Adjustement

Finally, we propose to study the impact of the "color adjustments". More precisely, we refer to color adjustment to denote all the processing operations aiming at improving the visual quality of colors by modifying, pixel-wise, the mapping between pixels value before and after processing ; typical examples of "color adjustments" processing includes highlight reconstruction, contrast enhancements, saturation modification, shadow and high exposition compensation, etc. ...[5]. We have used the Auto-Level tool throughout the preceding section which tunes the exposure parameters for each image, actually randomizing those parameters. A natural question is whether fixing the color adjustment parameters for an entire base lowers the intrinsic difficulty of the given base while not introducing inconsistency at the same time.

To answer this question we chose 6 different images from the BOSSBase and used RT's Auto-Level tool to get

[5] Those processing are referred to as "exposure correction" under RawTherapee software ; we, however, use the term "color adjustments" to avoid confusion with "exposure compensation".



(a) Image# 16 (b) Image# 17 (c) Image# 18 (d) Image# 7354 (e) Image# 7420 (f) Image# 7483

Figure 1. *The 6 images from BOSSbase on which the color adjustment parameters have been picked to set the constant toning on all the images..*

the color adjustment parameters of each specific image. For visualization, those images are presented in Figure 1. Those images have been manually selected because of their very different luminance histograms resulting in the setting of widely different parameters for color adjustment or color adjustment.

We then generated 6 databases by applying the specific color adjustment parameters to the entire BOSSBase, with all the other processing algorithms set to their default values. The Results are summarized in Table 7.

We repeated the experiment by applying the same color adjustment (or color adjustments) parameters but also using the Unsharp mask sharpening method with parameters defined as for **USM4** ; this allows to study the effect or color constant color adjustment and randomized adjustments on a more difficult development settings and, hence, to get a better sense of the interplay between two class of processing parameters. The results obtained from this experiment are summarized in Table 8.

Train\Test	AL	16	17	18	7354	7420	7483
AL	**0.224**	0.243	0.202	0.213	0.213	0.204	0.231
16	0.225	**0.260**	0.217	0.219	0.226	0.219	0.245
17	0.229	0.239	**0.208**	0.215	0.220	0.203	0.241
18	0.230	0.247	0.202	**0.239**	0.217	0.209	0.248
7354	0.225	0.234	0.206	0.205	**0.209**	0.200	0.225
7420	0.227	0.235	0.208	0.211	0.215	**0.208**	0.243
7483	0.222	0.237	0.198	0.209	0.219	0.208	**0.217**

Table 7. Intrinsic difficulty and consistence of images bases for different color adjustment parameters. Each row corresponds to a training base while each column corresponds to a testing base. Both training and testing base come from the BOSSBase. The header names correspond to the name of the image where the color adjustment parameters were taken.

The main conclusion on the impact of color adjustments that can be deduced from Table 7 and 8 are summarized below:

- In Table 7, we can note that fixing the color adjustment parameters may have a noticeable effect on the intrinsic difficulty, with a range of P_E of about 5%.
- Fixing the color adjustment parameters does not always lower the intrinsic difficulty, and may in fact increase the difficulty by a non-negligible amount (about 3.5% using settings from image # 16).

Train\Test	USM4	16	17	18	7354	7420	7483
USM4	**0.310**	0.332	0.302	0.311	0.310	0.294	0.321
16	0.332	**0.373**	0.307	0.342	0.325	0.320	0.357
17	0.333	0.323	**0.297**	0.311	0.322	0.292	0.355
18	0.341	0.352	0.313	**0.350**	0.323	0.310	0.350
7354	0.313	0.324	0.295	0.310	**0.308**	0.293	0.319
7420	0.319	0.331	0.300	0.312	0.313	**0.296**	0.321
7483	0.317	0.318	0.300	0.307	0.309	0.294	**0.310**

Table 8. Intrinsic difficulty and consistence of images bases for different color adjustment parameters with each base sharpened using USM4. Each row corresponds to a training base while each column corresponds to a testing base. Both training and testing base come from the BOSSBase. The header names correspond to the name of the image where the color adjustment parameters were taken.

- Inconsistency is not greatly introduced between bases not sharing the same exposure parameters as the difference from the intrinsic difficulty never exceed 3%.
- Interestingly, we note that some color adjustment settings may have a beneficial effect when used in the training phase (for instance, settings from image # 7483 improves classification with of # 16, # 17, # 18 with regard to their intrinsic difficulties) while having a rather high inconsistency when used as a testing base.
- Generally speaking, a rather small inconsistency is observed when trained on the dataset with toning automatically set by the software.

Thus setting proper color adjustment parameters can be a good strategy to lower the intrinsic difficulty of a difficult image base or, more realistically, to design a training base to classify a specific image.

Co-occurrence Analysis of Different Development Settings

In order to get a better interpretation of the inconsistency between different development settings, we decided to observe from a statistical point of view the impact of the development pipeline on joint distribution of neighboring pixels. By performing such an analysis we try to recast the CSM problem into a more theoretical perspective in order to see if the inconsistencies highlighted in the previous sections can be justified by the fact that the joint-distributions of pixels are very different after different development pipelines.

To do so we generated an artificial raw image in 14 bits coded DNG format, on this image each even column is equal to 8192 and each odd column to 5461. Then, we altered this image by adding Normal noise of standard deviation equal to 8.192 for even column and 5.461 for odd column. The goal here is to challenge both the demosaicing algorithms with fluctuating color components and denoising algorithms with noise distributed as the sensor noise [5, 19].

We develop the image in 8bits tif (uncompressed) format and compute the gray scale version by averaging the 3 color components. Each time we use the same methodology as in the previous sections, by only modifying one development setting.

We display the empirical co-occurrence between two diagonal neighboring pixels on Figure 2 for the different development settings of RawTherapee used for example in Table 10. We choose the 2D co-occurrence matrix because it is a good proxy for the SPAM [15] and SRM features [6], which are features that are very efficient in steganalysis.

It is interesting to notice that, starting with the very same raw image, each development leads to a very specific co-occurrence matrix. For example, the USM1 and USM2 settings present sparse distributions while RL3 and RL4 present dense distributions. Note that the symmetry is due to the fact that the columns alternate between two average values. We have also checked that these discrepancies are also verified for horizontal or vertical neighborhoods.

Another important remark is the fact that developments presenting similar co-occurrence matrices are the most consistent ones (compare distribution shapes and results of Table 10), this is the case for for example for the developments (b) to (f) or (k) to (o), that are, respectively, Unsharp Mask (USM) and Denoising (DEN). Similar detection performances also imply similar co-occurrences, see for example distributions (b) and (c) and lines or columns USM1 and USM2 of Table 10.

Further works are needed using other raw images and by considering other dependencies, but these experiments strengthen the idea that one important part the cover source mismatch is due to the development pipeline and its way to alter the statistical distribution of the image during the development process.

Conclusion

The present work proposes another original look at the Cover Source Mismatch phenomenon. While it has often been assumed that this issue arises because of the difference between camera models, using raw images we show that this effect is largely due to the processing applied between image acquisition and storage. We note that this study does not contradict previous works, each camera model is likely to use specific image processing algorithms, but we refine it. We study the effect of three main types of processing that are both the most widely used and the most influencing in terms of CSM: namely image sharpening, denoising and color adjustments. We show that application of two different image processing algorithms have a great influence on source mismatch; especially compared to the mismatch when using the same algorithm with different parameters. We also show that those processing algorithms can drastically change the difficulty of steganalysis on the generated dataset.

Table 9 sums up the principle conclusions of our analysis and try to answer to the difficult question "How to define a Source?" by recalling the impact of each parameter on the Inconsistency or the Difficulty. We can define the parameters defining a source as the set of parameters impacting these two parameters and from the our analysis, the main parameters are the JPEG Quality Factor and the Processing pipeline that have to be first considered, the software and the color adjustment setting can be taken into account if they are not used with their default settings. Finally our experiments suggest that the camera model or the ISO setting are not of prime importance.

To rephrase our conclusion, and contrary to what was proposed in [21], a general advise to fight again the Cover Source Mismatch should not be ’Toss that BOSSbase, Alice!’ but rather ’Keep that BOSSbase(-Raw), Alice, but take care of the development pipeline!’

Future works are necessary to study in more details the Cover Source Mismatch phenomenon and especially to understand why some processing have a large mismatch and some a much lower. Eventually, on steganalysis’ side, future works should seek at a method that would prevent suffering the “image processing” mismatch.

Inconsistency and Difficulty	Minor	Moderate	Major
JPEG QF			X
Camera sensor	X		
Software		X	
Processing			X
ISO	X		
Color Adjustment		X	

Table 9. How to define a Source ? : Comparison between the different parameters studied in the paper in term of Inconsistency and Difficulty.

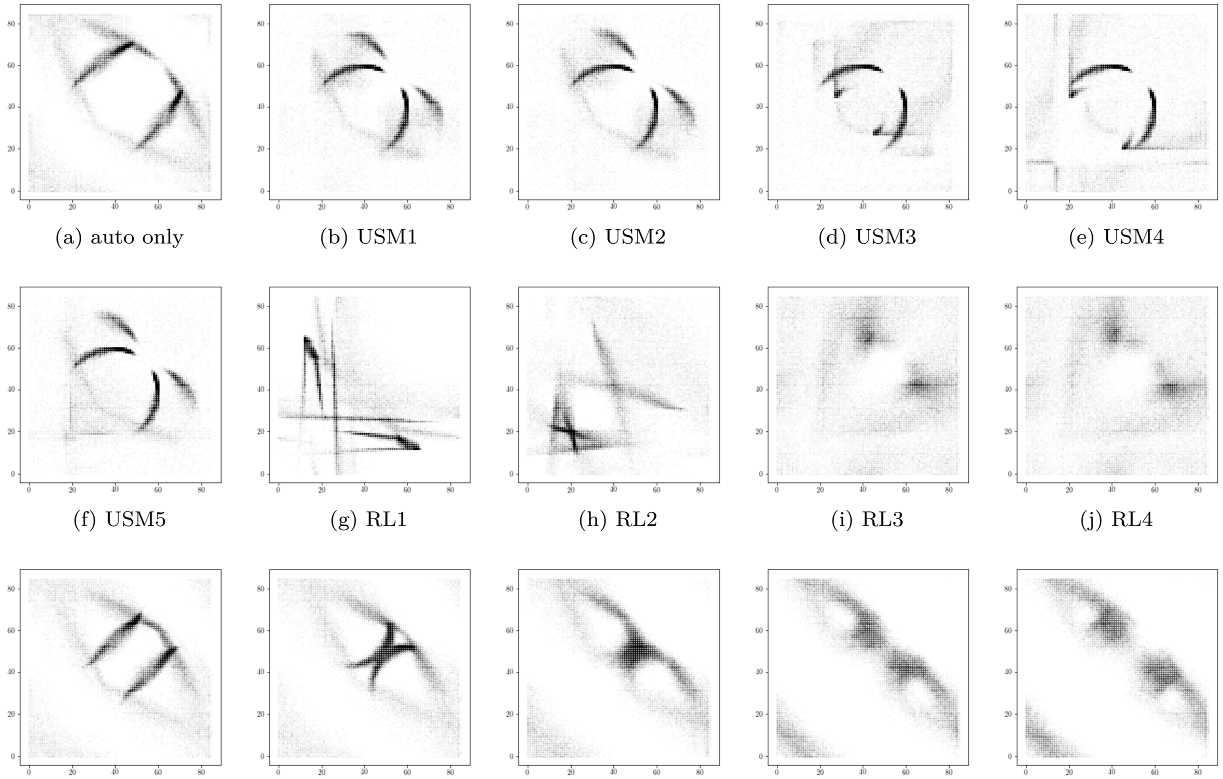
Acknowledgments

This work was partially supported by the French ANR DEFALS program (ANR-16-DEFA-0003).

Authors would like to thank Dr Dirk Borghys for pointing out that LightRoom and RawTherapee were not using the same JPEG quantization matrix at the same QF.

References

[1] P. Bas, T. Filler, and T. Pevný. Break our steganographic system — the ins and outs of organizing boss. In *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, pages 59–



(a) auto only (b) USM1 (c) USM2 (d) USM3 (e) USM4

(f) USM5 (g) RL1 (h) RL2 (i) RL3 (j) RL4

(k) DEN1 (l) DEN2 (m) DEN3 (n) DEN4 (o) DEN5

Figure 2. *Co-occurrences after different developments, dark values show populated area of the joint distribution.*

70, Prague, Czech Republic, May 18–20, 2011. LNCS vol.6958, Springer-Verlag, New York.

[2] G. Cancelli, G. Doerr, M. Barni, and I. Cox. A comparative study of ±1 steganalyzers. In *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, pages 791 –796, oct. 2008.

[3] R. Cogranne and J. Fridrich. Modeling and extending the ensemble classifier for steganalysis of digital images using hypothesis testing theory. *IEEE Transactions on Information Forensics and Security*, 10(12):2627–2642, 2015.

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

[5] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

[6] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.

[7] J. Fridrich, T. Pevnỳ, and J. Kodovskỳ. Statistically undetectable jpeg steganography: dead ends challenges, and opportunities. In *Proceedings of the 9th workshop on Multimedia & security*, pages 3–14. ACM, 2007.

[8] V. Holub and J. Fridrich. Low-complexity features for jpeg steganalysis using undecimated dct. *Information Forensics and Security, IEEE Transactions on*, 10(2):219–228, Feb 2015.

[9] V. Holub, J. Fridrich, and T. Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1):1–13, 2014.

[10] A. D. Ker, P. Bas, R. Böhme, R. Cogranne, S. Craver, T. Filler, J. Fridrich, and T. Pevnỳ. Moving steganography and steganalysis from the laboratory into the real world. In *Proceedings of the first ACM workshop on Information hiding and multimedia security*, pages 45–58. ACM, 2013.

[11] A. D. Ker and T. Pevnỳ. A mishmash of methods for mitigating the model mismatch mess. In *Media Watermarking, Security, and Forensics*, page 90280I, 2014.

[12] J. Kodovsky and J. Fridrich. Effect of image downsampling on steganographic security. *IEEE Transactions on Information Forensics and Security*, 9(5):752–762, 2014.

[13] J. Kodovskỳ, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *Information Forensics and Security, IEEE Transactions on*, 7(2):432–444, April 2012.

[14] J. Kodovskỳ, V. Sedighi, and J. J. Fridrich. Study of cover source mismatch in steganalysis and ways to mitigate its impact. In *Media Watermarking, Security, and Forensics*, page 90280J, 2014.

[15] T. Pevny, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security*, 5(2):215–224, 2010.

[16] V. Sedighi, R. Cogranne, and J. Fridrich. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 11(2):221–234, 2016.

[17] 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*, pages 15–23. ACM, 2015.

[18] H. C. D. B. Steganalysis and A. Westfeld. High capacity despite better steganalysis: F5–a steganographic algorithm. In *Information Hiding: 4th International Workshop, IH 2001, Pittsburgh, PA, USA, April 25-27, 2001. Proceedings*, volume 2137, page 289. Springer Science & Business Media, 2001.

[19] T. H. Thai, R. Cogranne, and F. Retraint. Statistical model of quantized dct coefficients: Application in the steganalysis of jsteg algorithm. *IEEE Transactions on Image Processing*, 23(5):1980–1993, 2014.

[20] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

[21] S. Vahid, J. Fridrich, and R. Cogranne. Toss that bossbase, alice! Society for Imaging Science and Technology, February 2016.

Author Biography

Quentin Giboulot is a Master student from Troyes University of Technology and is pursuing his research within his degree of engineering in System, Network and Telecommunication. He should graduate by spring 2018 and is willing to continue his research with a Ph.D. in steganography and steganalysis.

Rémi Cogranne holds the position of Associate Professor at Troyes University of Technology (UTT), France, since 2013. He had received his PhD in Systems Safety and Optimization in 2011 and his engineering degree in computer science and telecommunication in 2008 both from UTT. He has been a visiting scholar at Binghamton University for year between 2014 and 2017. During his studies, he took a semester off to teach in a primary school in Ziguinchor, Senegal and studied one semester at Jönköping University, Sweden. Since 2011, his work has been generously supported by various industrial and governmental contracts that lead to more than 55 papers and 3 International patents. His main research interests are in hypothesis testing with applications for image forensics, steganalysis and steganography and for computer network anomaly detection.

Patrick Bas received the Electrical Engineering degree from the Institut National Polytechnique de Grenoble, France, in 1997, and then the Ph.D. degree in signal and image processing from Institut National Polytechnique de Grenoble, France, in 2000. He has co-organized the 2nd Edition of the BOWS-2 contest on watermarking in 2007, and the first edition of the BOSS contest on steganalysis in 2010. From 2013 to 2016 Patrick Bas was associate editor of IEEE Transactions of Information Forensics and Security (IEEE TIFS). Patrick Bas is the current group leader of the team working on Signal and Images in the CRISTAL Lab.

Train\Test	AL	USM1	USM2	USM3	USM4	USM5	RL1	RL2	RL3	RL4	DEN1	DEN2	DEN3	DEN4	DEN5
AL	**0.231**	0.333	0.329	0.323	0.343	0.390	0.285	0.300	0.264	0.269	0.218	0.214	0.229	0.236	0.250
USM1	0.268	**0.318**	0.314	0.311	0.334	0.382	0.302	0.306	0.318	0.317	0.375	0.391	0.421	0.439	0.455
USM2	0.247	0.320	**0.315**	0.321	0.330	0.387	0.292	0.307	0.291	0.301	0.383	0.386	0.413	0.445	0.459
USM3	0.309	0.322	0.321	**0.316**	0.334	0.390	0.318	0.345	0.319	0.326	0.338	0.343	0.382	0.403	0.404
USM4	0.303	0.321	0.327	0.311	**0.335**	0.386	0.316	0.338	0.323	0.325	0.392	0.385	0.416	0.439	0.458
USM5	0.319	0.324	0.325	0.327	0.342	**0.390**	0.398	0.424	0.323	0.408	0.458	0.485	0.484	0.466	0.471
RL1	0.232	0.328	0.327	0.324	0.353	0.387	**0.291**	0.301	0.264	0.269	0.230	0.218	0.222	0.231	0.188
RL2	0.249	0.334	0.330	0.333	0.359	0.394	0.293	**0.286**	0.267	0.280	0.330	0.341	0.370	0.406	0.421
RL3	0.255	0.326	0.328	0.341	0.365	0.389	0.306	0.293	**0.261**	0.263	0.241	0.257	0.255	0.275	0.314
RL4	0.244	0.333	0.334	0.332	0.348	0.387	0.299	0.290	0.257	**0.264**	0.264	0.299	0.320	0.347	0.348
DEN1	0.288	0.349	0.351	0.351	0.385	0.405	0.315	0.333	0.304	0.314	**0.217**	0.198	0.185	0.188	0.194
DEN2	0.307	0.382	0.377	0.384	0.395	0.424	0.349	0.368	0.340	0.345	0.225	**0.191**	0.157	0.152	0.168
DEN3	0.310	0.391	0.385	0.396	0.404	0.429	0.360	0.381	0.336	0.342	0.234	0.202	**0.152**	0.128	0.118
DEN4	0.336	0.402	0.395	0.409	0.413	0.437	0.378	0.396	0.349	0.364	0.246	0.218	0.159	**0.123**	0.101
DEN5	0.358	0.414	0.403	0.408	0.421	0.433	0.379	0.414	0.377	0.377	0.258	0.219	0.165	0.126	**0.098**

Table 10. Intrinsic difficulty and consistence of images bases for different processing parameters. Each row corresponds to a training base while each column corresponds to a testing base. The training base is the BOSSBase with M9 images removed, the testing base is the M9Base-ISO160. Data hiding was carried with nsF5 at payload 0.04 bpac.

Train\Test	AL	USM1	USM2	USM3	USM4	USM5	RL1	RL2	RL3	RL4	DEN1	DEN2	DEN3	DEN4	DEN5
AL	**0.018**	-0.009	0.001	-0.006	-0.013	-0.000	0.003	0.020	0.015	0.002	-0.081	-0.110	-0.136	-0.148	-0.141
USM1	-0.017	**0.011**	0.007	-0.004	0.006	0.005	-0.001	-0.098	-0.030	-0.022	-0.048	-0.058	-0.058	-0.022	-0.017
USM2	-0.092	0.005	**0.003**	0.009	-0.002	0.003	-0.025	-0.005	-0.091	-0.060	-0.048	-0.069	-0.042	-0.016	-0.008
USM3	0.041	0.017	0.017	**-0.001**	0.004	0.015	0.013	0.036	0.018	0.022	-0.115	-0.127	-0.072	-0.060	-0.073
USM4	-0.070	0.003	0.011	-0.004	**0.007**	0.004	-0.045	-0.084	-0.046	-0.028	-0.057	-0.087	-0.043	-0.030	-0.015
USM5	-0.010	-0.009	0.008	0.004	0.018	**0.005**	0.035	0.049	-0.026	0.062	-0.010	0.000	0.008	-0.015	-0.002
RL1	0.001	0.009	0.013	0.013	0.018	0.001	**0.015**	0.006	-0.012	-0.015	-0.044	-0.102	-0.154	-0.134	-0.206
RL2	-0.053	0.011	0.003	-0.003	0.016	0.008	-0.019	**0.011**	-0.010	0.005	-0.078	-0.082	-0.084	-0.059	-0.051
RL3	0.001	-0.005	0.003	0.003	0.011	0.007	0.010	-0.004	**0.004**	-0.001	-0.124	-0.138	-0.181	-0.171	-0.147
RL4	-0.002	0.001	0.020	-0.008	0.001	-0.001	0.005	0.002	0.006	**-0.002**	-0.116	-0.095	-0.116	-0.109	-0.117
DEN1	0.015	-0.018	-0.016	-0.037	-0.006	-0.010	-0.011	0.014	0.014	-0.010	**0.018**	0.014	0.007	0.001	-0.035
DEN2	-0.012	-0.028	-0.025	-0.027	-0.026	-0.003	-0.021	0.013	0.010	0.004	0.017	**0.008**	-0.007	-0.003	-0.015
DEN3	-0.024	-0.037	-0.032	-0.030	-0.025	-0.004	-0.030	-0.007	-0.031	-0.028	0.015	0.017	**0.016**	0.012	0.019
DEN4	0.024	0.009	-0.003	0.019	0.010	0.005	0.024	-0.026	-0.006	-0.004	0.021	0.029	0.012	**0.009**	0.012
DEN5	0.008	-0.022	-0.018	-0.002	-0.003	-0.001	-0.028	0.028	0.008	-0.017	0.021	0.014	0.009	0.005	**0.006**

Table 11. Difference in P_E between training on the BOSSBase with M9 images removed and training on the same base as the testing set (here M9Base-ISO160) a negative result means a better classification when training on the BOSSBase while a positive result means better classification when training on the same base as the testing set. Data hiding was carried with nsF5 at payload 0.04 bpac.

Train\Test	AL	USM1	USM2	USM3	USM4	USM5	RL1	RL2	RL3	RL4	DEN1	DEN2	DEN3	DEN4	DEN5
AL	**0.327**	0.452	0.459	0.444	0.46	0.476	0.4	0.417	0.358	0.368	0.13	0.081	0.057	0.041	0.052
USM1	0.362	**0.426**	0.433	0.424	0.434	0.455	0.406	0.442	0.397	0.397	0.255	0.258	0.253	0.249	0.287
USM2	0.389	0.435	**0.434**	0.429	0.435	0.453	0.419	0.448	0.423	0.435	0.284	0.261	0.261	0.293	0.294
USM3	0.357	0.429	0.429	**0.419**	0.427	0.456	0.401	0.439	0.397	0.402	0.262	0.228	0.192	0.156	0.174
USM4	0.374	0.425	0.428	0.419	**0.43**	0.451	0.422	0.442	0.407	0.416	0.308	0.266	0.232	0.215	0.267
USM5	0.401	0.438	0.43	0.429	0.437	**0.456**	0.428	0.462	0.443	0.443	0.41	0.405	0.415	0.438	0.432
RL1	0.334	0.446	0.463	0.454	0.444	0.465	**0.402**	0.416	0.371	0.385	0.161	0.098	0.094	0.096	0.099
RL2	0.358	0.458	0.457	0.452	0.463	0.475	0.417	**0.413**	0.371	0.376	0.207	0.174	0.175	0.204	0.261
RL3	0.348	0.468	0.471	0.462	0.457	0.482	0.415	0.422	**0.361**	0.37	0.171	0.149	0.135	0.098	0.089
RL4	0.351	0.461	0.472	0.468	0.478	0.485	0.424	0.421	0.359	**0.364**	0.169	0.145	0.16	0.13	0.105
DEN1	0.36	0.481	0.481	0.477	0.476	0.489	0.442	0.43	0.401	0.399	**0.122**	0.078	0.032	0.029	0.048
DEN2	0.371	0.482	0.484	0.479	0.481	0.486	0.443	0.435	0.411	0.419	0.116	**0.067**	0.03	0.021	0.021
DEN3	0.377	0.481	0.482	0.479	0.484	0.489	0.448	0.437	0.422	0.426	0.127	0.069	**0.026**	0.014	0.01
DEN4	0.362	0.476	0.484	0.478	0.484	0.486	0.445	0.435	0.402	0.401	0.135	0.076	0.035	**0.014**	0.01
DEN5	0.384	0.483	0.484	0.483	0.484	0.489	0.448	0.453	0.41	0.426	0.143	0.085	0.031	0.02	**0.012**

Table 12. Intrinsic difficulty and consistence of images bases for different processing parameters. Each row corresponds to a training base while each column corresponds to a testing base. The training base is the BOSSBase with M9 images removed, the testing base is the M9Base-ISO160. Data hiding was carried with J-UNIWARD at payload 0.5 bpnzac.

Train\Test	AL	USM1	USM2	USM3	USM4	USM5	RL1	RL2	RL3	RL4	DEN1	DEN2	DEN3	DEN4	DEN5
AL	**0.009**	-0.009	-0.004	-0.005	-0.003	0.009	0	-0.002	-0.012	-0.007	-0.038	-0.043	-0.062	-0.144	-0.131
USM1	-0.028	**-0.009**	-0.004	-0.007	-0.005	0.001	-0.018	-0.02	-0.028	-0.027	-0.086	-0.094	-0.154	-0.176	-0.119
USM2	0.003	0.002	**0.012**	-0.004	-0.006	0.007	-0.004	-0.01	0.001	-0.023	-0.086	-0.117	-0.181	-0.149	-0.145
USM3	-0.012	0	-0.008	**0.002**	0	0.005	-0.032	-0.022	-0.022	-0.034	-0.103	-0.123	-0.171	-0.227	-0.232
USM4	-0.027	-0.006	-0.007	-0.003	**-0.011**	-0.001	-0.008	-0.013	-0.026	-0.029	-0.104	-0.172	-0.192	-0.21	-0.174
USM5	-0.032	-0.002	-0.008	-0.012	-0.002	**0.001**	-0.012	-0.001	0	-0.024	-0.066	-0.074	-0.071	-0.052	-0.059
RL1	0.009	0.008	0.025	0.015	0.003	0.001	**0.009**	-0.002	0.021	0.016	-0.043	-0.094	-0.033	-0.039	-0.108
RL2	-0.009	-0.009	-0.011	-0.016	-0.005	0	-0.014	**0.015**	-0.004	-0.003	-0.064	-0.116	-0.064	-0.083	-0.065
RL3	0.012	0.012	0.014	0.01	-0.021	0	0.014	0.009	**0.01**	0.007	-0.07	-0.021	-0.066	-0.12	-0.185
RL4	0.001	0.006	-0.001	0.015	0.002	0.016	-0.006	0.013	0.005	**0.002**	-0.03	-0.046	-0.009	-0.042	-0.154
DEN1	0.008	-0.004	-0.008	-0.003	-0.01	-0.002	0.003	-0.008	-0.004	-0.014	**0.018**	0.014	-0.008	-0.018	-0.02
DEN2	0.003	-0.005	-0.005	-0.007	-0.006	-0.007	-0.01	-0.015	0.001	0.003	0	**0.01**	0.005	0.004	0
DEN3	0.007	-0.007	-0.007	-0.006	-0.006	-0.002	0.001	-0.018	0.011	0.006	0.014	0.013	**0.003**	0.003	-0.001
DEN4	-0.022	-0.012	-0.006	-0.009	-0.006	-0.004	-0.013	-0.015	-0.016	-0.02	0.01	0.011	0.013	**0.005**	0.001
DEN5	-0.013	-0.007	-0.007	-0.006	-0.006	-0.003	-0.01	0.003	-0.011	0.001	-0.007	0.006	0.005	0.009	**0.003**

Table 13. Difference in P_E between training on the BOSSBase with M9 images removed and training on the same base as the testing set (here M9Base-ISO160) a negative results means a better classification when training on the BOSSBase while a positive result means better classification when training on the same base as the testing set. Data hiding was carried with J-UNIWARD at payload 0.5 bpnzac.

Train\Test	AL	USM1	USM2	USM3	USM4	USM5	RL1	RL2	RL3	RL4	DEN1	DEN2	DEN3	DEN4	DEN5
AL	**0.235**	0.332	0.328	0.324	0.347	0.389	0.288	0.300	0.260	0.269	0.219	0.234	0.240	0.231	0.272
USM1	0.257	**0.317**	0.323	0.318	0.337	0.383	0.299	0.316	0.298	0.303	0.394	0.395	0.425	0.449	0.461
USM2	0.256	0.322	**0.315**	0.317	0.333	0.383	0.294	0.301	0.286	0.289	0.368	0.389	0.429	0.445	0.460
USM3	0.309	0.325	0.319	**0.314**	0.336	0.383	0.325	0.349	0.334	0.322	0.344	0.327	0.381	0.400	0.419
USM4	0.303	0.327	0.329	0.314	**0.335**	0.385	0.316	0.334	0.324	0.329	0.386	0.392	0.406	0.420	0.441
USM5	0.308	0.330	0.324	0.327	0.351	**0.391**	0.319	0.344	0.330	0.339	0.466	0.460	0.466	0.469	0.465
RL1	0.227	0.327	0.324	0.321	0.348	0.387	**0.282**	0.298	0.258	0.286	0.237	0.229	0.237	0.214	0.248
RL2	0.275	0.338	0.331	0.336	0.353	0.391	0.291	**0.290**	0.268	0.277	0.312	0.341	0.382	0.398	0.384
RL3	0.244	0.335	0.327	0.342	0.358	0.389	0.303	0.300	**0.265**	0.266	0.256	0.257	0.258	0.255	0.268
RL4	0.258	0.339	0.325	0.347	0.363	0.392	0.307	0.291	0.256	**0.268**	0.264	0.280	0.322	0.293	0.320
DEN1	0.274	0.359	0.357	0.357	0.381	0.410	0.324	0.349	0.309	0.304	**0.218**	0.197	0.169	0.179	0.182
DEN2	0.298	0.379	0.373	0.383	0.399	0.421	0.341	0.367	0.337	0.338	0.224	**0.194**	0.161	0.151	0.159
DEN3	0.309	0.390	0.381	0.398	0.403	0.428	0.353	0.382	0.333	0.345	0.233	0.204	**0.150**	0.125	0.113
DEN4	0.348	0.402	0.400	0.415	0.414	0.432	0.378	0.388	0.366	0.366	0.247	0.209	0.163	**0.132**	0.103
DEN5	0.373	0.419	0.408	0.419	0.423	0.434	0.405	0.392	0.382	0.384	0.274	0.218	0.170	0.130	**0.100**

Table 14. Intrinsic difficulty and consistence of images bases for different processing parameters. Each row corresponds to a training base while each column corresponds to a testing base. The training base is the BOSSBase with M9 images removed, the testing base is the M9Base-ISO1250. Data hiding was carried with nsF5 at payload 0.04 bpac.

Train\Test	AL	USM1	USM2	USM3	USM4	USM5	RL1	RL2	RL3	RL4	DEN1	DEN2	DEN3	DEN4	DEN5
AL	**-0.005**	-0.005	-0.005	-0.016	-0.008	-0.006	-0.024	-0.001	-0.007	-0.004	-0.062	-0.075	-0.116	-0.162	-0.142
USM1	-0.066	**-0.023**	-0.020	-0.024	-0.024	-0.015	-0.041	-0.051	-0.031	0.014	0.042	-0.051	-0.038	0.003	0.003
USM2	-0.019	-0.014	**-0.025**	-0.029	-0.030	-0.016	-0.062	-0.013	-0.034	-0.004	-0.024	-0.056	-0.059	-0.021	-0.004
USM3	0.003	-0.014	-0.018	**-0.026**	-0.024	-0.011	0.004	-0.007	0.044	0.003	-0.030	-0.087	-0.086	-0.087	-0.036
USM4	-0.034	-0.007	-0.005	-0.027	**-0.013**	-0.011	-0.029	-0.036	0.001	0.006	0.015	-0.048	-0.078	-0.049	-0.019
USM5	-0.037	-0.019	-0.020	-0.034	-0.018	**-0.005**	-0.018	0.018	-0.025	-0.003	0.081	0.024	-0.023	-0.014	-0.011
RL1	-0.024	-0.006	-0.012	-0.025	-0.006	-0.010	**-0.020**	-0.008	-0.006	0.010	-0.014	-0.026	-0.059	-0.147	-0.166
RL2	-0.011	-0.005	-0.015	-0.012	-0.012	-0.004	-0.034	**-0.008**	-0.015	-0.009	-0.065	0.008	0.027	0.020	-0.046
RL3	-0.017	-0.005	-0.010	-0.008	-0.020	-0.009	-0.017	-0.003	**-0.001**	-0.012	-0.033	-0.082	-0.108	-0.170	-0.165
RL4	-0.016	-0.014	-0.022	-0.013	-0.008	-0.004	-0.013	-0.016	-0.014	**-0.008**	-0.036	0.001	0.022	-0.118	-0.088
DEN1	0.014	0.013	0.001	0.000	0.011	0.006	0.005	0.035	0.036	0.013	**0.019**	0.005	-0.027	-0.038	-0.047
DEN2	0.013	0.023	0.019	0.021	0.019	0.014	0.015	0.028	0.039	0.030	0.009	**0.004**	-0.006	-0.034	-0.062
DEN3	-0.015	-0.008	-0.011	0.001	-0.001	0.001	-0.027	0.011	-0.012	-0.016	0.005	0.009	**0.005**	-0.008	-0.036
DEN4	-0.029	-0.017	-0.013	0.004	-0.009	-0.007	-0.057	-0.028	-0.020	-0.021	-0.015	-0.006	0.007	**0.011**	-0.001
DEN5	-0.016	-0.008	-0.011	-0.012	-0.007	-0.008	-0.031	-0.050	-0.030	-0.031	0.006	-0.005	0.016	0.003	**0.001**

Table 15. Difference in P_E between training on the BOSSBase with M9 images removed and training on the same base as the testing set (here M9Base-ISO1250) a negative result means a better classification when training on the BOSSBase while a positive result means better classification when training on the same base as the testing set. Data hiding was carried with nsF5 at payload 0.04 bpac.