Study of Cover Source Mismatch in Steganalysis and Ways to Mitigate its Impact

Jan Kodovský, Vahid Sedighi, and Jessica Fridrich

Department of ECE, Binghamton University, NY, USA

ABSTRACT

When a steganalysis detector trained on one cover source is applied to images from a different source, generally the detection error increases due to the mismatch between both sources. In steganography, this situation is recognized as the so-called cover source mismatch (CSM). The drop in detection accuracy depends on many factors, including the properties of both sources, the detector construction, the feature space used to represent the covers, and the steganographic algorithm. Although well recognized as the single most important factor negatively affecting the performance of steganalyzers in practice, the CSM received surprisingly little attention from researchers. One of the reasons for this is the diversity with which the CSM can manifest. On a series of experiments in the spatial and JPEG domains, we refute some of the common misconceptions that the severity of the CSM is tied to the feature dimensionality or their "fragility." The CSM impact on detection appears too difficult to predict due to the effect of complex dependencies among the features. We also investigate ways to mitigate the negative effect of the CSM using simple measures, such as by enlarging the diversity of the training set (training on a mixture of sources) and by employing a bank of detectors trained on multiple different sources and testing on a detector trained on the closest source.

Keywords: Steganalysis, steganography, cover source mismatch, machine learning

1. INTRODUCTION

The problem of the so-called cover source mismatch (CSM) pertains to the situation when a steganalyzer is trained on a source of images and then presented with cover (or stego) examples from a different source. The mismatched detector typically exhibits a lower detection accuracy. The negative impact of the CSM has already been commented upon in [3, 7] but the problem became more widely recognized and documented after the BOSS competition [1] in the work of BOSS participants [6, 9]. The drop of performance due to CSM can range from a small to moderate loss of detection accuracy to literally catastrophic results when a relatively accurate detector on one source completely fails on another.

In general, the impact of the CSM on detection accuracy depends on many factors, including the steganographic algorithm and the distribution of the payload size, the steganalysis feature space, the properties of the cover source, and the actual implementation of the steganalyzer. The differences between cover sources can accept many different forms, examples of which are images of different resolution, size, and format, processed images, images acquired by different hardware, and under different conditions. Given the great diversity and the associated complexity of the problem, it is not surprising that, at present, the problem of the CSM is rather poorly understood. It has also been identified as one of the main obstacles when deploying steganalysis in real world [11].

This article is rather exploratory in nature. Positioning ourselves somewhere between the lab and the real world, we intentionally constrained our experiments to controlled image sets in order to eliminate any hidden variables and to better isolate the effects of various aspects of the CSM on detection accuracy. The focus is on gaining some understanding of how various factors contribute to the severity of the CSM and whether its impact can be mitigated using selected simple measures. On examples, we debunk some widely believed misconceptions that the features' high dimensionality or their "fragility"* are the main reasons for a catastrophic

E-mail: {jan.kodovsky,vsedigh1,fridrich}@binghamton.edu; `http://dde.binghamton.edu`

*See the definition of fragility at the end of Section 4.4.

loss of detection performance under CSM. Evidence gathered through experiments suggests that the interplay (mutual dependencies) among features are equally important factors affecting the ability of the detector to generalize to previously unseen sources of images.

The experiments are divided into two parts based on the steganographic embedding domain. In each domain, we investigate one embedding algorithm using two feature sets – one low-dimensional set and one rich model. The steganalysis detectors are always implemented as binary classifiers using the ensemble classifier [16]. In the spatial domain, the sources differ by the image-acquiring hardware and applied processing. In the JPEG domain, we restrict our experiments to the CSM due to different quantization tables. We investigate the severity of the CSM impact based on the feature dimension, type of calibration (in the JPEG domain), and processing applied to the cover source. We also study the effectiveness of two simple counter-measures: training a single classifier on a mixture of sources and training a bank of classifiers first and then testing a given unseen source on the closest source used for training. The detection accuracy is measured using the total minimum probability of error under equal priors averaged over ten realizations of each experiment (over various random splits of the training and testing sets).

In the next section, we summarize several different approaches for fighting the CSM that were proposed in the machine learning community and in steganalysis. The next two sections contain the main results of this paper. Section 3 focuses on experiments in the spatial domain, while Section 4 deals with the steganalysis of JPEG images. Each section contains a detailed interpretation of the results and a summary of the lessons learned. The paper is concluded in Section 5, where we outline possible avenues for future research.

2. CSM IN STEGANALYSIS: PRIOR ART

The problem of a mismatched detector is not new. It has been extensively studied within the context of robust statistical hypothesis testing [22]. Such robust methods guarantee an upper bound on the drop of detection accuracy as a function of some measure quantifying the mismatch between the assumed and true distributions. On the side of the machine learning community, the interest in constructing robust classifiers has been steadily increasing. The methods of domain adaptation [8, 10] can be applied when the analyst has access to a set of (unlabeled) samples from the testing source and can utilize this information for the training of the detector[†]. This requires a potentially expensive retraining of the classifier for each new testing domain. Domain generalization, on the other hand, tries to transform the feature space so that the differences between a multitude of training sources are as small as possible while keeping the ability of the transformed features to distinguish between the classes (of cover and stego images). A general domain generalization method recently developed is the Domain Independent Component Analysis (DICA) [18]. The term transfer learning [2, 19] is used for general techniques directed at addressing the problem when there is a mismatch between the distributions of training and testing data.

For domain adaptation and generalization techniques to work, it is necessary that the training and testing sources be close. This condition is, however, hardly satisfied in real world applications in steganalysis, at least with current feature representations of digital media. Techniques with a lesser tendency to overtrain to a specific training cover source include the Constrained Least Squares (CLS) [21], which appears to be a special case of DICA with linear kernels in both the example and label spaces.[‡] The CLS is an example of a promising direction based on the idea of linearly transforming the feature space to minimize the statistical spread of cover features while maximizing the correlation between the stego features and the embedding change rate.

The effect of the classifier complexity, together with the size and diversity of the training set has been experimentally studied in [17]. There, the authors concluded that simpler classifiers, such as the perceptron or the ensemble classifier [16], indeed appear to produce more robust detectors than more complex machine learning tools, such as Gaussian SVMs. The authors also pointed out that it is better to train on samples from a larger number of diverse sources than on a less diverse source.

[†]This situation corresponds to the setup of the BOSS competition [1].

[‡]T. Pevný. Personal communication, November 2013.

	Cover source	Format/processing	Native resolution
1	Canon EOS 400D	RAW, `dcraw`	3906×2602
2	CanonEOS 7D	RAW, `dcraw`	5202×3465
3	Canon EOS Digital Rebel XSi	RAW, `dcraw`	4290×2856
4	Pentax K20D	RAW, `dcraw`	4688×3124
5	Nikon D70	RAW, `dcraw`	3039×2014
6	Leica M9	RAW, `dcraw`	5216×3472
7	Canon EOS 5D Mark_II	RAW, `dcraw`	5634×3753
8	Canon EOS 20D	RAW, `dcraw`	3522×2384
9	Canon EOS 550D	RAW, `dcraw`	5202×3465
10	Canon 6D	RAW, `ufraw`	5496×3670
11	Canon 6D	JPEG	5472×3648
12	Sony DSC HX 100V	JPEG	4608×3456
13	Leica M9	RAW, `dcraw`, resized to 1024×1024	5216×3472

Table 1. Thirteen cover sources for experiments in the spatial domain.

3. SPATIAL DOMAIN

3.1 Cover sources

For experiments in the spatial domain, we selected 13 different sources listed in Table 1, each consisting of 10,000 images. Sources 1–9 were prepared using 9 different cameras by converting 1,000 images originally acquired at full resolution in the raw format to a 24-bit color TIFF image using '`dcraw`'[§] (calling '`dcraw -d -T img_filename`') and then converting the true-color image to 8-bit grayscale in Matlab (`rgb2gray`). Ten 512×512 disjoint image blocks were cut from each of these images to form 10,000 images per source. Source 10 was prepared in the same manner except the '`dcraw`' routine was replaced with '`ufraw`'[¶] (calling '`ufraw-batch img_filename`'). Sources 11–12 were produced by taking 1,000 images directly in the JPEG format in the camera and decompressing them to the spatial domain. Canon 6D images were compressed with two JPEG quality factors for the luminance channel: 75 and ≈98, while the SONY used three customized quantization tables. Two tables were close to the standard table with quality factor ≈90 while the other was close to 94. Source 13 was obtained from 5,000 raw Leica M9 images (a different batch than those used for Source 6) using the same conversion as for Source 6. The grayscale full-size images were then resized in Matlab with the bicubic kernel with no antialiasing so that the smaller dimension was 1024, centrally cropped to 1024×1024, and finally cut up into two 512×512 images positioned in the lower part.

3.2 Stego algorithm and feature sets

All tests in this section were carried out with non-adaptive LSB matching with a fixed change rate of 0.1 changes per pixel. The steganalyzer was the ensemble ver. 2.0 (`http://dde.binghamton.edu/download/ensemble/`) run with the default settings. The experiments were conducted with two different feature vectors: 1) the 338-dimensional SQUARE submodel of the Spatial Rich Model (SRM) [5] with the quantization step $q = 1$, and 2) the 12,753-dimensional SRMQ1 feature vector.

3.3 Experiments

The experiments consisted of the following tasks. First, we trained on Source i and tested on Source j to see the impact of the CSM on detection accuracy. For training, a randomly-selected half of the 10,000 cover-stego pairs was chosen, while the testing was done on 5,000 randomly selected pairs from Source j. For $i = j$, the no CSM case, care was taken to make sure the training and testing sets were disjoint.

Second, we tested three simple strategies for mitigating the CSM impact. The first, called 'Mixture', consisted of training on a mixture of 5,000 images from 12 sources (12×417 cover-stego pairs uniformly randomly selected

[§] `http://www.cybercom.net/~dcoffin/dcraw/`

[¶] `http://ufraw.sourceforge.net/`

from each source) and testing on 5,000 images from the remaining 13th source. The second strategy, 'Closest source', used a bank of 13 classifiers trained on one randomly selected half of each source. Given a set of images from testing Source j, we first determined the closest source among the other 12 sources and then tested images from Source j using the classifier trained for the closest cover source. In the third method, 'Closest source (indiv.)', we steganalyzed each testing image separately with the classifier trained for the cover source closest to that individual image. Note that the first and third methods do not require other examples from the testing source than the test image, while the second method assumes the availability of multiple images from the testing source. These images are not used for retraining but merely for computing the distances between the sources.

We experimented with several different measures of source closeness. The overall best performer was a simple L_2 norm between the centers of gravity of cover feature clusters. The Mahalanobis distance performed slightly worse, while a measure of closeness based on the detection error of a classifier trained to distinguish both cover sources was unable to provide useful information when the sources were very different and the classifier error was near zero (e.g., between any raw source and the decompressed JPEGs).

		SQUARE (338)					
	Cover source	No CSM	Worst case	Median	Mixture	Closest source	Closest source (indiv.)
1	Canon EOS 400D	.0457 ± .0010	.4992 ± .0002	.1043 ± .0026	.0863 ± .0018	.0610 ± .0017	.1123 ± .0013
2	CanonEOS 7D	.2529 ± .0027	.5009 ± .0011	.3625 ± .0044	.3623 ± .0047	.3615 ± .0039	.3590 ± .0026
3	Canon EOS Digital Rebel XSi	.0774 ± .0013	.4998 ± .0001	.1745 ± .0057	.1479 ± .0029	.1104 ± .0101	.1327 ± .0058
4	Pentax K20D	.0869 ± .0016	.4991 ± .0002	.1338 ± .0027	.1271 ± .0016	.1042 ± .0018	.1611 ± .0024
5	Nikon D70	.0816 ± .0022	.4995 ± .0003	.1825 ± .0035	.1674 ± .0034	.1180 ± .0038	.1355 ± .0031
6	Leica M9	.0720 ± .0016	.4991 ± .0002	.2238 ± .0057	.1758 ± .0031	.1416 ± .0022	.2256 ± .0027
7	Canon EOS 5D Mark_II	.0389 ± .0008	.4984 ± .0003	.0780 ± .0040	.0465 ± .0009	.0536 ± .0017	.2429 ± .0006
8	Canon EOS 20D	.0465 ± .0011	.4985 ± .0003	.0947 ± .0046	.0726 ± .0017	.0567 ± .0015	.0992 ± .0013
9	Canon EOS 550D	.0742 ± .0020	.4994 ± .0002	.1300 ± .0035	.1161 ± .0029	.0969 ± .0025	.1319 ± .0018
10	Canon 6D	.1209 ± .0022	.4992 ± .0003	.4683 ± .0157	.4740 ± .0086	.4941 ± .0005	.4578 ± .0120
11	Canon 6D (JPEG)	.0028 ± .0007	.4702 ± .0024	.4381 ± .0098	.0834 ± .0150	.3029 ± .0089	.2437 ± .0094
12	Sony DSC HX 100V (JPEG)	.0000 ± .0000	.4963 ± .0010	.4878 ± .0035	.0593 ± .0237	.0002 ± .0001	.1180 ± .0108
13	Leica M9 (Resized)	.0619 ± .0015	.4928 ± .0029	.3400 ± .0085	.2366 ± .0116	.3325 ± .0055	.4105 ± .0038

Table 2. Detection error for no CSM, for CSM (worst and median values when training on the remaining 12 sources), and when employing one of the three strategies. See the text for more details. Feature space: 338-dimensional 'SQUARE'.

		SRM (12,753)					
	Cover source	No CSM	Worst case	Median	Mixture	Closest source	Closest source (indiv.)
1	Canon EOS 400D	.0022 ± .0004	.4987 ± .0002	.0116 ± .0010	.0081 ± .0007	.0104 ± .0008	.0440 ± .0008
2	CanonEOS 7D	.0348 ± .0012	.4983 ± .0003	.2349 ± .0130	.1303 ± .0102	.4744 ± .0059	.2913 ± .0036
3	Canon EOS Digital Rebel XSi	.0037 ± .0005	.4994 ± .0001	.0260 ± .0023	.0214 ± .0034	.0136 ± .0011	.0726 ± .0018
4	Pentax K20D	.0067 ± .0006	.4978 ± .0003	.0372 ± .0018	.0185 ± .0010	.0228 ± .0009	.0310 ± .0005
5	Nikon D70	.0052 ± .0006	.4986 ± .0002	.0248 ± .0009	.0210 ± .0025	.0229 ± .0012	.0420 ± .0013
6	Leica M9	.0063 ± .0005	.4976 ± .0009	.0667 ± .0070	.0468 ± .0043	.0641 ± .0022	.0672 ± .0033
7	Canon EOS 5D Mark_II	.0062 ± .0007	.4976 ± .0003	.0197 ± .0009	.0107 ± .0011	.0113 ± .0005	.0438 ± .0003
8	Canon EOS 20D	.0033 ± .0006	.4967 ± .0003	.0134 ± .0020	.0080 ± .0007	.0091 ± .0008	.0295 ± .0002
9	Canon EOS 550D	.0058 ± .0006	.4986 ± .0002	.0222 ± .0029	.0127 ± .0011	.0174 ± .0010	.0285 ± .0006
10	Canon 6D	.0402 ± .0017	.4994 ± .0002	.4991 ± .0002	.4749 ± .0084	.4988 ± .0004	.4988 ± .0001
11	Canon 6D (JPEG)	.0029 ± .0005	.4696 ± .0106	.3511 ± .0162	.0812 ± .0134	.3487 ± .0019	.3156 ± .0090
12	Sony DSC HX 100V (JPEG)	.0000 ± .0000	.4948 ± .0018	.2929 ± .0362	.1297 ± .0494	.0037 ± .0027	.1804 ± .0313
13	Leica M9 (Resized)	.0516 ± .0011	.4845 ± .0014	.4089 ± .0126	.2797 ± .0045	.4845 ± .0014	.4407 ± .0081

Table 3. Detection error for no CSM, for CSM (worst and median values when training on the remaining 12th sources), and when employing one of the three strategies. See the text for more details. Feature space: 12,753-dimensional 'SRMQ1'.

The results of all experiments are summarized in Tables 2–3 showing the minimum total detection error under

equal priors,

$$P_{\rm E} = \min_{P_{\rm FA}} (P_{\rm FA} + P_{\rm MD})/2, \tag{1}$$

averaged over ten splits of the training and testing sources into halves (over random selection of images for the mixture) and the statistical spread expressed using the Mean Absolute Deviation (MAD). The first column concerns the case of no CSM. The second and third columns show $P_{\rm E}$ for the worst case of the CSM and the median $P_{\rm E}$ over all 12 training sources different than the testing source. These two columns are meant to show how bad the impact of the CSM can be. Columns 4–6 show the detection error $P_{\rm E}$ when employing one of the three strategies to mitigate the CSM impact.

For sources that are not very diverse (1–9), the three simple strategies seem to be quite effective in suppressing the negative impact of the CSM on detection, which can be quite bad (c.f., columns 2 ans 3). Surprisingly, this remains true when steganalyzing with the compact 'SQUARE' feature space as well as the rich model 'SRMQ1'. The main differences between Sources 1–9 are due to different sensor resolution, which affects the correlations among neighboring pixels) and processing during the signal transfer and quantization. The remainder of the processing pipeline stayed the same – it was executed using '`dcraw`'.

The three simple strategies failed for Sources 10–13. Source 10 differs from the first 9 sources in the RAW format converter – '`ufraw`' was used instead of '`dcraw`'. The differences in both processing routines apparently produce very different sources. Although untested, we hypothesize that sources generated using either of the two raw convertors sensitively depend on the settings, which include the color interpolation algorithm as well as the parameters for further processing, such as color correction, white balance, gamma correction, noise reduction, and a multitude of other processing that can be used to enhance the final image, which includes adjustment of shadows, highlights, blacks, whites, contrast, exposure, tint, color, saturation, vibrance, clarity, denoising, lens-distortion correction and chromatic aberration (both involve resampling). Given the enormous diversity a raw camera image can be processed, the three simple strategies can only be effective in practice when appropriately scaled up.

It is worth pointing out the results for the two decompressed JPEG sources no. 11 and 12. Since $\approx 70\%$ of Canon 6D images were compressed with quality factor ≈ 98, this source is a good training source for Source 12, which contains decompressed JPEGs from SONY with quality factors ≈ 90 and ≈ 94. On the other hand, the SONY images are relatively poor training data for Canon 6D decompressed JPEGs, which contain images originally stored with a much lower quality factor of 75. This very limited experiment seems to suggest that cover sources comprised of decompressed JPEGs could be effectively steganalyzed by one of the tested strategies by enlarging the range of quality factors for the training sources.

The last source, Leica M9 resized to 1024×1024 (and cropped to 512×512) is again an outlier source because none of the 12 training sources contains images with resizing artifacts. As shown in [15], the resizing algorithm and its parameters can have a very substantial effect on steganalysis.

4. JPEG DOMAIN

Since JPEG compression is a type of a low-pass filter, it largely suppresses differences among sources acquired using different cameras as well as differences due to processing, such as resizing. On the other hand, JPEG images depend on a vector parameter – the quantization table(s). Since quantization has a dramatic effect on the distribution of the DCT coefficients forming the JPEG file, it also has a major effect on the accuracy with which steganography can be detected and on the robustness of the detectors to the CSM. Therefore, in this section we study the impact of the CSM caused by mismatched quantization tables. Note that it is not feasible to construct a detector for each quantization table as many cameras today use custom tables that may even depend on the image content.

4.1 Cover sources

We use the BOSSbase 1.01 database consisting of 10,000 8-bit grayscale images of size 512×512. Multiple different sources were created by compressing this mother database with quantization tables for JPEG quality factors $65 - 100$, as well as a set of custom quantization tables extracted from JPEG images coming from several different camera models.

4.2 Stego algorithm and feature sets

All tests were carried out with the steganographic algorithm nsF5 [4]. The stego images were obtained by embedding a fixed relative payload of 0.1 bits per nonzero AC DCT coefficient (bpac). The steganalyzer was again the ensemble 2.0 run with default settings. To see the impact of feature complexity, we worked with the 548-dimensional CC-PEV feature vector [13, 20] and the 22,510-dimensional CC-JRM rich model [14]. We also investigated the effect of calibrating by difference (CD). In calibration by difference, the feature space has half the dimensionality of its CC version as we always subtract from each feature its reference.

4.3 Experiments with the CC-PEV feature vector

In all experiments, one half of the images (5,000) was always used for training, and the other half for testing. Care was taken to make sure that the training and testing sets never contained the same image in either cover or stego form.

Similar to the experiments in the spatial domain, in Table 4 we show the detection error P_{E} for eight testing sources (listed in the first column) when training on the same quantization table (column 2), on a source of JPEG images with quality factors 75 and 90 (columns 3 and 4), when training on a uniform mixture of images compressed with quality factors $65, 67, 69, \ldots, 99$ (column 5), and when training on a quantization table closest to the testing table (shown in parenthesis). The closeness was determined using the following weighted measure:

$$d\left(Q^{\mathrm{trn}}, Q^{\mathrm{tst}}\right) = \sum_{k,l=0}^{7} \left(\frac{Q^{\mathrm{trn}}_{kl} - Q^{\mathrm{tst}}_{kl}}{Q^{\mathrm{tst}}_{kl}} \right)^2, \tag{2}$$

where Q^{trn} and Q^{tst} stand for the training and testing 8×8 quantization tables, respectively.

	CC-PEV (548)				
Cover source	no CSM	QF 75	QF 90	Mixture	Closest QT
Canon S2 IS-1	.1696 ± .0027	.4122 ± .0334	.3415 ± .0241	.2094 ± .0018	.2154 ± .0058 (98)
Canon S2 IS-4	.2892 ± .0022	.3165 ± .0073	.4021 ± .0201	.2934 ± .0020	.2797 ± .0025 (73)
Canon EOS 6D-1	.1669 ± .0023	.4205 ± .0189	.3225 ± .0244	.2103 ± .0032	.2162 ± .0083 (98)
Nokia-1 (standard 85)	.2568 ± .0024	.2862 ± .0163	.3298 ± .0305	.2575 ± .0019	.2479 ± .0032 (85)
Panasonic DMC FZ50-1	.1818 ± .0026	.4421 ± .0122	.3808 ± .0178	.2272 ± .0103	.2937 ± .0186 (99)
Panasonic DMC FZ50-7	.2274 ± .0021	.4002 ± .0373	.3109 ± .0147	.2397 ± .0040	.2175 ± .0036 (96)
Sony-1 (standard 95)	.2169 ± .0022	.3930 ± .0350	.3422 ± .0108	.2262 ± .0041	.2109 ± .0052 (95)
Sony-4	.2289 ± .0030	.3958 ± .0357	.2570 ± .0130	.2355 ± .0033	.2443 ± .0053 (94)

Table 4. Study of the impact of the CSM due to quantization table (taken from the corresponding camera model) for the CC-PEV (548) feature vector. The integer following the dash in the source name identifies different quantization tables for each camera.

The columns for QF 75 and QF 90 nicely illustrate how important this research direction is. Training on a mixture works reasonably well to mitigate the negative effect of the CSM. Experiments with different mixtures of quality factors, including a denser mixture containing all quality factors between 65 and 100, did not produce consistently better results than what is reported in the table. The 'Closest QT' strategy also works well.

Figure 1 shows an alternative visualization of all results of Table 4 in one graph. The x-axis corresponds to the 'no CSM' error while the y-axis shows all other scenarios (QF75, QF90, Mixture, Closest QT). The diagonal solid line corresponds to the no-CSM benchmark. The farther the points from the diagonal are, the more severe the effect of the CSM is. This graph nicely shows that QF 75 and QF 90 are bad strategies, while 'Closest QT' and 'Mixture' stay close to the diagonal. The individual points correspond to individual cameras.

In Figure 2 (top), we show the detection error as a function of the training quality factor to better see the impact of the CSM. We do so only for Sony-4 as a representative example since qualitatively similar conclusions

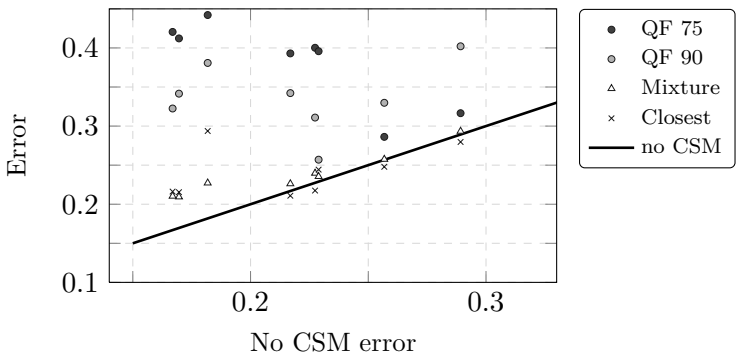

Figure 1. Alternative visualization of Table 4.

can be made for other tested cameras. The solid horizontal line indicates the no-CSM performance, the dashed horizontal line corresponds to the mixture, while the dots correspond to the performance when training on individual standard quality factors – this includes the cases of QF 75, QF 90, and QF 94 (identified as the closest). Error bars represent the MAD values over 10 training/testing splits. The line with crosses shows the distance measure (2) (transformed through the logarithm, scaled and shifted). Note that the closest standard table corresponds to quality factor 94. The distance also positively correlates with the curve representing individual errors.

4.3.1 Effect of calibration

Figure 2 (bottom) shows the detection errors for the PEV feature vector calibrated by difference (CD). Note that the 274 CD-PEV features have a significantly lower statistical spread over different splits into the training / testing sets.

Table 5 is the CD-PEV counterpart of Table 4 (the last two columns are copied for convenience). By comparing the 'Mixture' strategy for both feature sets, one can see that it pays off to use the CC-PEV feature set, even though it is more prone to the CSM than the CD-PEV.

	CD-PEV (274)					CC-PEV (548)	
Cover source	no CSM	QF 75	QF 90	Mixture	Closest QT	Mixture	Closest QT
Canon S2 IS-1	.2191 ± .0017	.3917 ± .0071	.3054 ± .0022	.2425 ± .0025	.2327 ± .0019 (98)	.2094 ± .0018	.2154 ± .0058 (98)
Canon S2 IS-4	.2929 ± .0013	.2944 ± .0017	.3326 ± .0050	.3182 ± .0020	.2941 ± .0017 (73)	.2934 ± .0020	.2797 ± .0025 (73)
Canon EOS 6D-1	.2157 ± .0022	.3956 ± .0071	.3081 ± .0032	.2357 ± .0013	.2291 ± .0011 (98)	.2103 ± .0032	.2162 ± .0083 (98)
Nokia-1	.2759 ± .0017	.2899 ± .0012	.2943 ± .0029	.2934 ± .0022	.2769 ± .0008 (85)	.2575 ± .0019	.2479 ± .0032 (85)
Panasonic DMC FZ50-1	.2132 ± .0016	.4239 ± .0067	.3555 ± .0037	.2374 ± .0025	.2381 ± .0034 (99)	.2272 ± .0103	.2937 ± .0186 (99)
Panasonic DMC FZ50-7	.2387 ± .0019	.3286 ± .0044	.2567 ± .0026	.2602 ± .0025	.2604 ± .0030 (96)	.2397 ± .0040	.2175 ± .0036 (96)
Sony-1	.2394 ± .0019	.3290 ± .0050	.2524 ± .0035	.2582 ± .0029	.2491 ± .0029 (95)	.2262 ± .0041	.2109 ± .0052 (95)
Sony-4	.2512 ± .0022	.3253 ± .0048	.2562 ± .0024	.2644 ± .0029	.2625 ± .0023 (94)	.2355 ± .0033	.2443 ± .0053 (94)

Table 5. Detection errors using CD-PEV (274) features. The last two columns are taken from Table 4 for reference.

4.4 Experiments with the CC-JRM feature vector

In this section, we report the results of the same experiments as in the previous section, except now the JRM is used instead of the PEV feature vector. Due to the computational complexity associated with extracting the 22,510-dimensional JRM feature vector, Table 6 shows the results only for four selected sources.

Even though the CC-JRM delivers better results than CC-PEV in the case of no CSM, under CSM the results are disastrous. The CC-JRM is extremely sensitive to the CSM, and neither 'Mixture' nor the 'Closest QT' strategy seem to help.

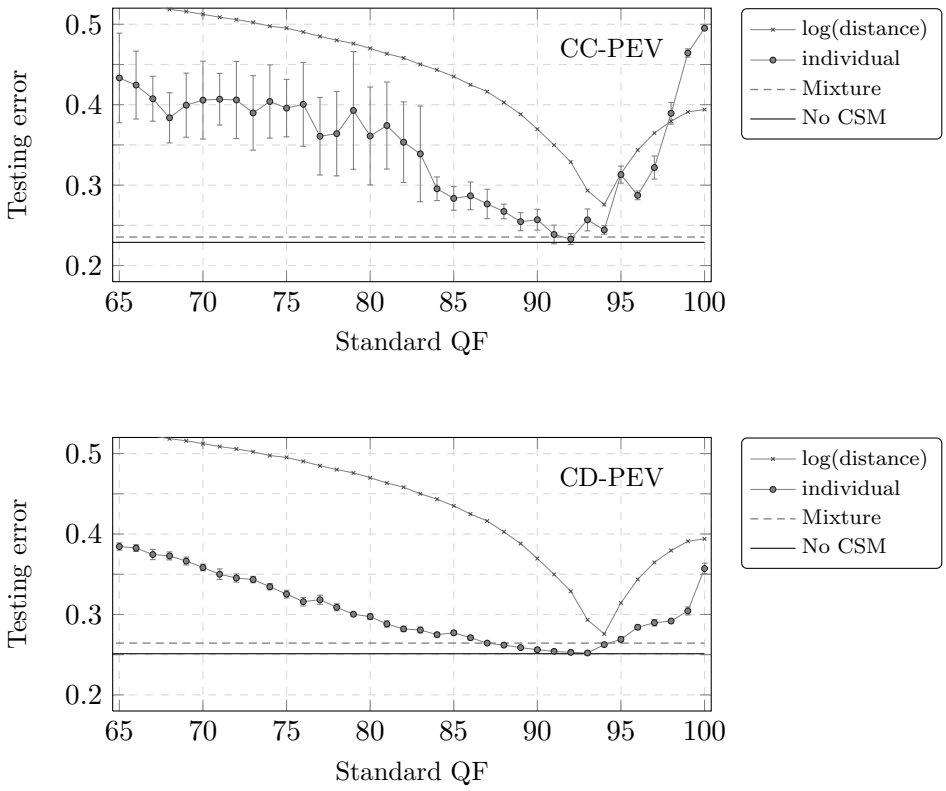

Figure 2. Detection error as a function of the training quality factor for Sony-4 and the CC-PEV (top) and the CD-PEV (bottom) feature vectors.

	CC-JRM (22,510)					CC-PEV (548)		
Cover source	no CSM	QF 75	QF 90	Mixture	Closest QT	no CSM	Mixture	Closest QT
Canon S2 IS-1	.1210 ± .0033	.4738 ± .0176	.4966 ± .0043	.4132 ± .0221	.4666 ± .0060 (98)	.1696 ± .0027	.2094 ± .0018	.2154 ± .0058 (98)
Canon EOS 6D-1	.1184 ± .0032	.4384 ± .0198	.4730 ± .0224	.4107 ± .0202	.4597 ± .0074 (98)	.1669 ± .0023	.2103 ± .0032	.2162 ± .0083 (98)
Panasonic DMC FZ50-7	.1428 ± .0020	.4775 ± .0137	.4759 ± .0125	.3316 ± .0226	.4992 ± .0004 (96)	.2274 ± .0021	.2397 ± .0040	.2175 ± .0036 (96)
Sony-4	.1546 ± .0025	.4652 ± .0131	.4807 ± .0065	.3589 ± .0151	.4776 ± .0045 (94)	.2289 ± .0030	.2355 ± .0033	.2443 ± .0053 (94)

Table 6. Steganalysis using CC-JRM (22,510) features. The last three columns are taken from Table 4 for reference.

In an effort to explain this intriguing phenomenon, we first decided to investigate the effect of calibration. In particular, we repeated the experiment with the 11,255-dimensional JRM calibrated by difference (CD-JRM). Then, we modified the ensemble classifier to always select the random subspaces of features so that a feature is always selected with its reference to not lose the power the reference features provide (CC-JRM-pairs).

Since the qualitative behavior appeared rather similar across the cover sources, in Table 7 we only report the results for the first testing source Canon S2 IS-1.

It seems that Cartesian calibration is not a good choice when dealing with the CSM and rich models. While the difference-calibrated CD-JRM has a worse no-CSM performance, it is rather resistant to the CSM, and both 'Mixture' and 'Closest QT' strategies work well. However, under the CSM the performance of the CD-JRM is comparable to much more compact CC-PEV.

The CC-JRM-pairs has the same performance as CC-JRM when there is no CSM. Under a CSM, it seems that preserving original-reference feature pairs in random subspaces helps a little, but not substantially. Calibrating by difference is much better. This can be explained by the fact that all features in the CC-JRM are referencing each other and thus both CC-JRM and CC-JRM-pairs cases are quantitatively similar.

Feature space	no CSM	QF 75	QF 90	Mixture	Closest QT (98)
CC-PEV (548)	.1696 ± .0027	.4122 ± .0334	.3415 ± .0241	.2094 ± .0018	.2154 ± .0058
CC-JRM (22,510)	.1210 ± .0033	.4738 ± .0176	.4966 ± .0043	.4132 ± .0221	.4666 ± .0060
CC-JRM-pairs (22,510)	.1204 ± .0033	.4790 ± .0099	.4965 ± .0033	.3617 ± .0337	.4446 ± .0145
CD-JRM (11,255)	.2083 ± .0035	.3462 ± .0055	.2776 ± .0058	.2091 ± .0023	.2118 ± .0061

Table 7. Detection error for three versions of the JRM: CC-JRM, CC-JRM with random subspaces selected in pairs feature-reference (CC-JRM-pairs), and the difference calibrated JRM (CD-JRM). The first row for CC-PEV is copied from Table 4 for comparison.

The "folklore" says that the CSM can be exacerbated by feature dimensionality or feature "fragility." By fragility, we mean the fact that in JRM, the majority of the features (co-occurrence bins) are collected separately for each DCT mode. This makes them less populated and also their distribution is much more sensitive to the cover source. In contrast, the co-occurrences in the CC-PEV feature vector are collected over all DCT modes and spatial directions, and thus are better populated and less sensitive to the cover source. It is thus tempting to attribute the much more robust behavior of the CC-PEV feature vector to its lower dimension and more robust structure (integral character). Unfortunately, both hypotheses are false as the next section shows, where we dissect the CC-PEV feature vector.

Index range	Feature type
1–11	Global histogram
12–66	Local histograms
67–165	Dual histograms
166	Variation
167–169	Blockiness
169–193	Inter-block 5×5 co-occurrences
194–274	Markov features (intra-block co-occurrences)
275–548	Reference features

Table 8. The structure of the CC-PEV feature vector

4.5 Dissecting the CC-PEV feature vector

We decided to test individual parts of the CC-PEV feature vector for their resistance to the CSM w.r.t. the quantization table. The CC-PEV feature vector [20] has the following components. The 81 Markov features are obtained as a sum of four horizontal, vertical, diagonal, and minor diagonal 9×9 co-occurrences of differences of absolute values of DCT coefficients (the co-occurrences are always taken in the direction of the difference). The inter-block co-occurrences are formed from DCT coefficients and not their absolute values as in the JRM.

In all tests below, we used the features indicated by the index range with their corresponding references. Thus, the feature dimension was twice as large as indicated in Table 8. Table 9 shows the testing error when training with different feature subsets of the CC-PEV for the testing table Canon S2 IS-1 whose closest standard quantization table is for quality factor 98 (the standard table exactly matches 27 quantization steps and all steps corresponding to spatial frequencies $k + l \leq 5$, $0 \leq k, l \leq 7$).

Perhaps, one of the most interesting aspects of the results is that, just like the rich model, the histogram features, Markov features, as well as the inter-block co-occurrences exhibit a quite large increase of the detection error due to the CSM. And this is despite the fact that all three sets are well-populated, low-dimensional, and definitely not fragile. The loss of performance improves only when the inter-block and Markov features are merged. Note that merging the histogram features with either inter-block or Markov features does not affect the impact of the CSM.

Strangely and quite inexplicably, the feature subsets in rows 2–4 and 6 do not achieve the lowest detection error for the closest quantization table.

Feature subset	no CSM	QF 75	QF 90	Closest QT (98)
1. Entire CC-PEV (1–274)	.1694±.0025	.3838±.0365	.3421±.0287	.2132±.0054
2. Markov (194–274)	.2723±.0012	.4307±.0071	.3814±.0037	.4099±.0059
3. Inter-block cooc (169–193)	.2454±.0024	.3122±.0102	.2773±.0028	.4634±.0025
4. Histogram features (1–165)	.2846±.0033	.4591±.0029	.4221±.0020	.4746±.0015
5. Inter + Markov (169–274)	.1948±.0025	.3541±.0063	.2959±.0046	.2683±.0086
6. Histogram + Inter (1–165,169–193)	.2028±.0018	.3110±.0153	.2772±.0120	.4097±.0067
7. Histogram + Markov (1–165,194–274)	.2032±.0023	.4394±.0119	.4070±.0126	.4004±.0081

Table 9. Detection error of selected feature subsets of the CC-PEV feature vector for the no CSM case and when training on a fixed QF of 75, 90, and the quantization table corresponding to the quality factor closest to the quantization table of the testing source (Canon S2 IS-1).

This experiment shows that, even for low-dimensional feature spaces, it is nearly impossible to predict which features will provide a robust detection and which will be quite susceptible to the CSM. The CSM severity changes abruptly and does not correlate with how well a given feature space detects embedding when there is no CSM (see, e.g., row 6). Apparently, the CSM impact is determined by complex dependencies among all features. From this point of view, approaches based on transforming the feature space while minimizing the features' covariance across covers (e.g., the CLS) hold some promise as a general methodology for mitigating the impact of the CSM.

5. CONCLUSIONS AND FUTURE WORK

The loss of detection power due to a mismatched detector complicates successful deployment of steganalysis tools in real world applications. Detectors designed to accurately identify steganographic embedding in certain sources may exhibit a very different performance when applied to images coming from a different source, examples of which the detector has never seen before. Preventing a detector from being overtrained to a specific training source is rather difficult due to the great diversity and complexity of typical digital media. With certainty, steganalysts are facing a trade-off between the performance on a matched source and the drop of detection accuracy for slightly different sources. One will likely have to sacrifice the detector's accuracy on a fixed source for increased robustness of the detector.

The sensitivity of a detector to the cover source mismatch in general depends on many factors. It has been hypothesized in the past that high-dimensional rich media models are "fragile" and overly sensitive, making them less suitable for real life applications. Based on the findings of this paper, however, even low-dimensional feature spaces of "integral character" can exhibit a catastrophic loss of performance due to even a small mismatch in the cover source.

In this paper, we study the effects of a CSM on the steganographic algorithm nsF5 in JPEG sources caused by different quantization tables. Simple strategies, such as training on images compressed with a mixture of quality factors or building a bank of detectors parametrized by the quality factor and then testing on the one with the closest table appear to work quite well for detectors built using the ensemble and the CC-PEV feature vector. Replacing the CC-PEV model with the CC-JRM model, however, leads to a catastrophic loss of detection accuracy even for slightly mismatched quantization tables. When replacing the Cartesian calibration by calibration by difference, robustness of the classifier is restored but at the cost of losing accuracy to the level of the much more compact CC-PEV space.

Experiments in the spatial domain were executed on non-adaptive LSM matching and on 13 different sources, including never compressed images, decompressed JPEGs, and resized images. For sources that have gone through a similar processing pipeline, the CSM impact could potentially by alleviated by increasing the diversity of the training set or by using a simple preclassifier and testing on the detector built for the closest training cover source. To make these approaches effective in real world applications, however, they would have to be substantially scaled up, making the steganalyzer potentially rather expensive to build.

The role of the CSM in steganalysis in general heavily depends on the application scenario and the information available to the steganalyst. These factors also determine the most appropriate approach to reveal the usage of

steganography. For example, there might be situations in practice when the steganalyst has a good knowledge of the cover source and can adjust the detector accordingly, essentially eliminating the CSM. Under these conditions, implementing the steganalyzer as a binary classifier is justified. In another scenario, one can be analyzing images posted on a social network to identify the "guilty player" rather than individual stego images. Here, unsupervised clustering techniques based on identifying outliers [12] may provide a better alternative than binary classifiers. Probably the hardest situation for analysis occurs when analyzing traffic in which each image comes from a different user and thus probably a different source ("a JPEG in the wild"). In this situation, the CSM constitutes a serious problem for successful deployment of steganalysis.

6. ACKNOWLEDGMENTS

The work on this paper was supported by Air Force Office of Scientific Research under the research grant number FA9950-12-1-0124. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of AFOSR or the U.S. Government.

REFERENCES

1. P. Bas, T. Filler, and T. Pevný. Break our steganographic system – the ins and outs of organizing BOSS. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, volume 6958 of Lecture Notes in Computer Science, pages 59–70, Prague, Czech Republic, May 18–20, 2011.
2. S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.
3. G. Cancelli, G. Doërr, I. J. Cox, and M. Barni. A comparative study of ±1 steganalyzers. In *Proceedings IEEE International Workshop on Multimedia Signal Processing*, pages 791–796, Cairns, Australia, October 8–10, 2008.
4. J. Fridrich and T. Filler. Practical methods for minimizing embedding impact in steganography. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 02–03, San Jose, CA, January 29–February 1, 2007.
5. J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2011.
6. J. Fridrich, J. Kodovský, M. Goljan, and V. Holub. Breaking HUGO – the process discovery. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 85–101, Prague, Czech Republic, May 18–20, 2011.
7. M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 1–13, San Jose, CA, January 16–19, 2006.
8. B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, June 16–21, 2012.
9. G. Gül and F. Kurugollu. A new methodology in steganalysis : Breaking highly undetactable steganograpy (HUGO). In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 71–84, Prague, Czech Republic, May 18–20, 2011.
10. J. Jing. A Literature Survey on Domain Adaptation of Statistical Classifiers. Available from: http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/survey/da_survey.html, March 2008.
11. A. D. Ker, P. Bas, R. Böhme, R. Cogranne, S. Craver, T. Filler, J. Fridrich, and T. Pevný. Moving steganography and steganalysis from the laboratory into the real world. In W. Puech, M. Chaumont, J. Dittmann, and P. Campisi, editors, *1st ACM IH&MMSec. Workshop*, Montpellier, France, June 17–19, 2013.
12. A. D. Ker and T. Pevný. Identifying a steganographer in realistic and heterogeneous data sets. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2014*, volume 8303, pages 0N 1–13, San Francisco, CA, February 3–5, 2012.

13. J. Kodovský and J. Fridrich. Calibration revisited. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 63–74, Princeton, NJ, September 7–8, 2009.
14. J. Kodovský and J. Fridrich. Steganalysis of JPEG images using rich models. In A. Alattar, N. D. Memon, and E. J. Delp, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2012*, volume 8303, pages 0A 1–13, San Francisco, CA, January 23–26, 2012.
15. J. Kodovský and J. Fridrich. Steganalysis in resized images. In *Proc. of IEEE ICASSP*, Vancouver, Canada, May 26–31, 2013.
16. J. Kodovský, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444, 2012.
17. I. Lubenko and A. D. Ker. Steganalysis with mismatched covers: Do simple classifiers help. In J. Dittmann, S. Katzenbeisser, and S. Craver, editors, *Proc. 13th ACM Workshop on Multimedia and Security*, pages 11–18, Coventry, UK, September 6–7 2012.
18. K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *Proceeding of the 30th International Conference on Machine Learning (ICML 2013)*, Atlanta, GA, June 16–21 2013.
19. S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, October 2010.
20. T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 3 1–14, San Jose, CA, January 29–February 1, 2007.
21. T. Pevný and A. D. Ker. A mishmash of methods for mitigating the model mismatch mess. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2014*, volume 9028, San Francisco, CA, February 3–5, 2014.
22. R. R. Wilcox. *Introduction to Robust Estimation and Hypothesis Testing, Third Edition (Statistical Modeling and Decision Science)*. Elsevier, 2012.