Facing the Cover-Source Mismatch on JPHide using Training-Set Design

Dirk Borghys, Patrick Bas, Helena Bruyninckx

Facing the Cover-Source Mismatch on JPHide using Training-Set Design

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

Patrick Bas
CNRS, École Centralle de Lille, Univ. of Lille, CRIStAL Lab
Lille
Patrick.Bas@centralelille.fr

Helena Bruyninckx
Royal Military Academy, Dept. of Mathematics
Brussels
Helena.Bruyninckx@rma.ac.be

ABSTRACT

This short paper investigates the influence of the image processing pipeline (IPP) on the cover-source mismatch (CSM) for the popular JPHide steganographic scheme. We propose to deal with CSM by combining a forensics and a steganalysis approach. A multi-classifier is first trained to identify the IPP, and secondly a specific training set is designed to train a targeted classifier for steganalysis purposes. We show that the forensic step is immune to the steganographic embedding. The proposed IPP-informed steganalysis outperforms classical strategies based on training on a mixture of sources and we show that it can provide results close to a detector specifically trained on the appropriate source.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**; **Malware and its mitigation**;

KEYWORDS

Digital image steganalysis, JPEG domain, cover-source mismatch, image processing pipeline, forensics-aware steganalysis

ACM Reference Format:
Dirk Borghys, Patrick Bas, and Helena Bruyninckx. 2018. Facing the Cover-Source Mismatch on JPHide using Training-Set Design. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

For digital images, machine learning based steganalysis is currently the methodology that achieves the best performances in a controlled environment, i.e. whenever the steganographic scheme, the payload size and the image source are known. However if one of these three parameters remains unknown, the performance of the steganalysis scheme can be jeopardized. The problem of Cover-Source Mismatch (CSM) occurs when the analyzed image sources are unknown. To the best of our knowledge, it has been identified in 2008 by the pioneering works of Cancelli *et al.* [5] and confirmed during the BOSS contest [2]. It states that a mismatch can occur and consequently degrades the classification performances if the source of the testing set is different from the source of the training set. As an example, during the BOSS contest and when applying the detectors optimized on BossBase (processed in a specific manner), to "real-world" images, it was noted that the performance dramatically drops.

Note that the CSM effect may be particularly observable when the training set comes from BossBase images. Indeed, if this database enables to compare steganographic and steganalysis results, its development pipeline is however extremely formatted: RAW images are first transformed into spatial (ppm) images using the free software DCraw with specific parameters. The images are then rescaled such that the largest dimension was 512 pixels and converted to greyscale, and for JPEG steganalysis the ppm images are compressed using standard quantization tables. All operations are performed using the 'convert' Unix command, which is very restrictive.

If the CSM can easily be observed, accurate characterization of the source in the literature is not straightforward. The term 'source' has been coined in 2011 [2, 6] and became an important topic in steganalysis research from 2012 onwards (e.g. [7–9, 11, 13, 14, 16]), definitions of a source are diverse and stay informal.

Ker and Pevny characterize a source as an actor [9], i.e. one user uploading a set of images on his social network account. The authors provide options to mitigate the CSM by normalizing independently the features of each user. Pasquet [16] *et al.* consider a source as a cluster of features, and they combine unsupervised and supervised training to conduct steganalysis. Finally, Kodovsky *et al.* [13] proposed to deal with different sources (here cameras) by training on a mixture of images coming from different sources and Lubenko and Ker [14] proposed to adopt a similar strategy on millions of images using a simple on-line classifier.

Recently Giboulot *et al.* [7] conducted an investigation to characterize the set of parameters that specify a 'source'. This paper considered the case were RAW images were acquired using various cameras and developed to JPEG images using photographic development softwares. The impact of the choice of camera, the acquisition parameters and the image processing pipeline (IPP) were considered. The paper showed that the acquisition parameters (including the camera type) have only a minor impact, but that the image processing parameters as well was the quantization table have the largest impact. The investigated processing parameters were sharpening, denoising, color adjustments and the choice of the development software.

The current paper follows the same methodology as [7] while investigating some complementary development pipelines such as white-balancing and demosaicing. It focuses on the popular JPHide embedding scheme. This scheme has been selected because the embedding is fast, and its detectability has already been analyzed within the CSM paradigm by Ker and Pevny [9]. Moreover, this

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

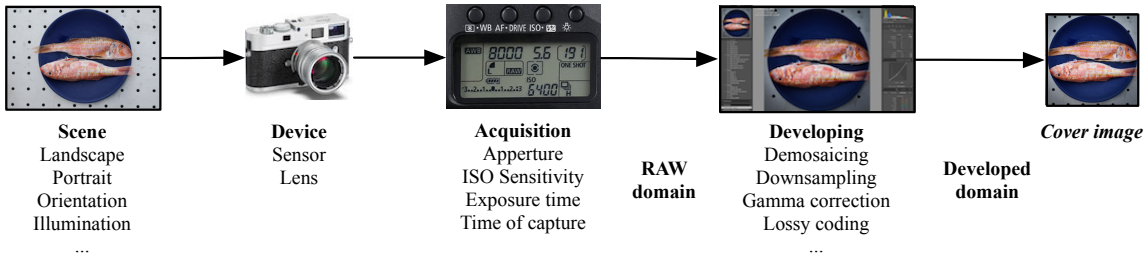

Figure 1: Pipeline of the cover image generation process which can be decomposed into four main steps (scene, device, acquisition, developing) representing parameters of the whole process.

work enables to highlight complementary conclusions with respect to [7].

A strategy to mitigate the impact of the CSM due to the image processing pipeline is also proposed. This strategy consists first of a forensics analysis by using steganalysis features to identify a processing pipeline similar to the one applied on the test image. Similarly to [7, 15], we show that the best detection results are obtained when the training is performed on an image database coming from a source similar to the one generating the test images.

1.1 The Image Processing Pipeline

A source can be defined w.r.t. the image generation process depicted in Figure 1 which shows that the creation of a cover image is linked to the succession of different parameters represented by (1) the scene that is captured, (2) the device which is used, (3) the acquisition settings used during the capture and (4) the developing step.

Each parameter is linked with a set of sub-parameters. The scene fluctuates according to the subject, but also according to the illumination or the orientation of the camera. The device is composed mainly of two elements: the sensor (which can be CMOS, CCD; color or monochrome) and the lens.

The acquisition phase relies on three parameters originating from the device: the lens aperture, the ISO sensitivity and the exposure time and one parameter which is the time of capture.

Finally, the developing step which is studied in this paper, contains a lot of processing steps and we list here the most important ones:

- the white-balance is a color transform needed to adjust to human perception of color under different illuminations,
- Gamma correction is a sample-wise transform which maps to a different tone,
- the demosaicing or Color Filter Array (CFA) interpolation step predicts two missing color components for each pixel from neighboring photo-site values,
- and the user can also apply other image processing operations such as denoising or sharpening.

2 METHODOLOGY

The methodology explored in this paper for reducing the CSM (cf. Figure 2) consists in first determining the image processing pipeline (IPP) that was applied to an image, and then exploiting that information for building an adequate training set that is used to train the steganalysis detectors. First, a multi-classifier is trained to identify the closest development process among a set of predefined ones. This operation is possible by extracting features from databases specifically developed from a database of RAW images. Secondly, the closest database of cover images is used as a training database for steganalysis by generating a corresponding set of stego images. Finally, a classical steganalysis methodology is applied by extracting features and training a classifier which is afterward used on the test image. Note that such a methodology is an example of forensics-aided steganalysis which was already briefly explored by Barni *et al.* [1] for distinguishing camera images from computer generated images before performing steganalysis.

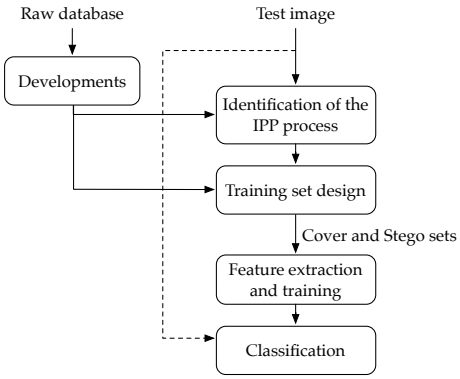

Figure 2: Schematic overview of the examined methodology.

3 EXPERIMENTAL SETUP

We chose to study a specific camera (the Leica M9) and in order to have a complete control of the image processing pipeline, the RAW images available in the BossBase are used in our experiments. In the original BossBase 2758 RAW images of the Leica M9 are available. These are used for creating test images corresponding to different choices of the processing parameters. After processing, the 2758 images are cropped into non-overlapping 512x512 images, resulting in a total of over 160000 images. This procedure thus allows an artificial increase of the image database. The selected cropping method results in a very high variation in scene content between different sub-images but the local statistics caused by the processing pipeline are the same as for the full-sized images.

Three photographic development tools were used in the experiment: the open-source software DCraw v9.25 (denoted "DC" in this paper) and RawTherapee v4.1.0 (http://rawtherapee.com/) ("RT") and the commercial software Adobe Lightroom© v6.0 ("LR6").

The three softwares were used for converting the RAW images into color JPEG images with a standard quantization table with quality factor 100 (STD100) for DCraw and RawTherapee and the Adobe Level 12 quantization table (Adobe12) for Lightroom. A 4:2:2 color sub-sampling was used in the JPEG conversion.

The steganographic method used in the current investigation is JPHide.

As mentioned in Giboulot *et al.*[7] the IPP in general modifies the content details of a picture and, hence, the resulting number of non-zero AC coefficients (nzac). In order to perform a fair comparison between the photographic developments performed by different IPPs, the payload size should remain constant over the different developments. Contrary to [7], note that we did not use the same message length for all images. The image contents of the different crops is so diverse that we decided to fix the message length for each of the crops but to keep it constant over the different developments applied to each cropped image. The message length is set to 10% of the nzac in the images developed by a specific processing chain (i.e. DCraw using a bilinear CFA interpolation). However, we believe that using a constant payload through all images and developments or using a constant payload only through developments should lead to similar conclusions.

For each of the investigated processing parameters, stego/cover pairs were created for the 160000 cropped images.

The used steganalytic detector is the Ensemble Classifier (EC) [12] based on the CC-JRM feature set [10] and used in a clairvoyant scenario, i.e. both the steganographic method and the embedding rate are known to the steganalyst. The false alarm, missed-detection and total error probabilities, (P_{fa}, P_{md} and $P_E = (P_{fa} + P_{md})/2$) were considered as performance metrics. In order to estimate these performance metrics, 10000 cover/stego image pairs were randomly selected from the set of 160000 images for training the detector. Another (disjoint) set of 10000 image pairs was used for validation. This was repeated five times in order to obtain an average value and a standard deviation for the performance metrics. For compactness, the paper reports only the average values for P_E.

4 IMPACT OF IMAGE PROCESSING PARAMETERS

Because Giboulot *et al.* [7] already investigated many image processing parameters, the current paper focuses on parameters not yet examined in that paper. These include white balancing, gamma correction, CFA demosaicing and the choice of the development software. For investigating the impact of white balancing and gamma correction, DCraw was used.

4.1 Impact of white balancing

The choice of the white balancing method influences the color appearance of an image. By default, DCraw uses a fixed white balance based on a color chart illuminated with a standard D65 lamp (cf. user manual of DCraw), which roughly corresponds to the average midday light in Northern/Western Europe.

Besides this default white balancing (WBdef) method, DCraw also allows to select two other types of white balance: camera (WBcam) and average (WBave). In WBcam the white balance is defined by the camera. In practice the photographer can choose between automatic white balancing (AWB) or a number of preset values depending on the lighting conditions (e.g. sunset, clear sky, clouded sky, ...). Each choice determines a color temperature applied in the white balancing [4].

In WBave the white balance is calculated by averaging over the complete image.

The white balance is thus partly defined by the camera during image acquisition, but can be overridden by the development software.

In the current experiment DCraw is used with all of its parameters set to their default value. Only the white balancing method is varied. The three available white balancing methods are applied and compared.

Figure 3 shows the results obtained for P_E for a steganalysis detector trained on images created by one of the three white balancing modes of DCraw (shown on the left of the table) and applied to images created by each of the three methods (top of the table). A colormap is assigned to the values for an easier visualization of the mismatch. The values on the diagonal correspond to the matched case and represent the "intrinsic difficulty" of the considered source [7].

The figure shows that the largest mismatch if found when training on WBdef and applying the trained detector to any of the two other methods. Training on WBdef and applying to the two other modes results in a more than tenfold increase of P_E. The mismatch between the two other modes is much milder.

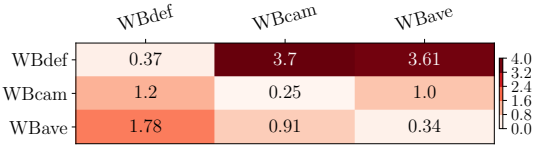

Figure 3: Influence of white balancing (P_E in %).

4.2 Impact of gamma correction

While [7] examines a range of manual tone adjustments in their investigation of the dependence on color adjustment, the current paper focuses on the more automatic process of gamma correction (GC).

In DCraw four gamma correction methods are available: BT-709, Adobe, ProPhoto and sRGB. BT709 is the default GC method in both DCraw and RawTherapee. Details on the various gamma correction methods and their parameters can be found in [18, 19]. The impact of gamma correction on the CSM is illustrated in figure 4. The largest mismatch is found between the default GC (GCdef) and the three other methods. Training on GCdef and applying the trained detector to the images created using the other three GC methods leads again to a more than tenfold increase of P_E w.r.t. the respective fully-matched cases. Between the three other methods the relative increase in P_E is between 1.3 (for training on GCsrgb images and

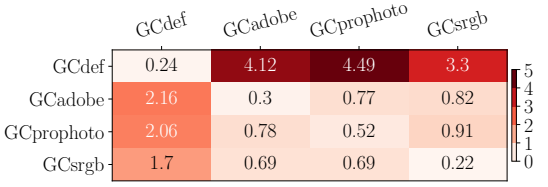

Figure 4: Influence of gamma correction (P_E in %).

testing on GCprophoto) and 4.1 (for training on GCprophoto and testing on GCsrgb). GCprophoto exhibits a higher intrinsic difficulty (i.e. the value on the diagonal corresponding to the fully-matched case) than the three other methods.

4.3 Impact of CFA interpolation and development software

For the conversion from RAW image to JPEG in this paper three development softwares are used: DCraw (DC), RawTherapee (RT) and Adobe Lightroom 6.0 (LR6). The first step in the conversion from RAW is the CFA interpolation (demosaicing). In this section the combined effect of the CFA interpolation and the choice of development software is investigated. Eleven IPPs were defined and examined. The results of the mismatch between them is presented in figure 5. Details about the eleven IPPs are given below. DCraw and RawTherapee were used with all parameters set to their default values. Only the demosaicing method was varied. For DCraw the (bi)linear, AHD, VNG and PPG interpolations were used (resp. denoted DClin, DCahd, DCvng, DCppg on the figure's axes). For RawTherapee AHD, Amaze and IGV were applied (RTahd, RTamaze and RTigv).

LR6 uses a proprietary demosaicing method for which no detailed documentation is available. For LR6 we considered four processing pipelines. In LR6Def all parameters were left to their default value. LR6 uses its own quantization tables (QTs) and in particular the QT at highest quality (level 12) differs from the standard table at quality factor QF=100. Therefore we also created LR6 images with the standard QT at QF=100 (LR6Std), consistent with the one used in the two other softwares. The LR6Std images were created using LR6 with all parameters set to their default value, but by exporting TIFF images instead of JPEG. The TIFF images are then converted to JPEG using the Python PIL library. The only difference between LR6Def en LR6Std is thus the QT.

LR6 performs several operations by default (denoising, sharpening, etc.). The default sharpening in LR6 is set to "level 25". For LR6 we also applied two other sharpening methods: a rather extreme sharpening in the development module at "level 125" (LR6DS125) and the "standard screen" (LR6SScr) method in the export module.

Figure 5 shows the results of the study of the mismatch between the various demosaicing methods and choice of the development softwares.

The figure clearly exhibits a block structure corresponding to the three softwares. The largest mismatch is thus obtained between different development softwares. Note that in [7] the development software was found to have only moderate impact. The different result we observe here is probably due to the different choice of the feature set; DCTR in [7] versus CC-JRM in the current paper. We intend to examine this further.

The difference in QT seems to be an important factor in the mismatch found between LR6 and DCraw. LR6Std shows indeed a much smaller mismatch w.r.t. DCraw than LR6Def.

The intrinsic difficulty of a source is lowest for images processed by DCraw and much higher for images processed by RT and LR6. The LR6DS125 images have the highest intrinsic difficulty. The authors think this is caused by the highly non-linear character of the sharpening applied in LR6 which leads to a high variability within the corresponding training set.

The largest relative increase of P_E due to a mismatch in demosaicing method (within the same development software) is between 2.2 (for training on RTigv and testing on RTamaze) and 4.8 (for training on DCvng and testing on DClin).

5 IMAGE PROCESSING PIPELINE (IPP) CLASSIFIER

The fact that steganalysis results depend on the IPP suggests that the used steganalysis feature set is sensitive to this IPP. We have therefore investigated whether it is possible to use the same feature set for detecting the IPP. Several papers have been published showing the usefulness of steganalysis features for digital image forensics and in particular for detecting image manipulations [3, 17].

The ensemble classifier (EC) yielding excellent results for steganalysis using large feature sets, in the current paper the EC is also used for constructing a supervised classifier of the IPP.

For assigning one of the N examined IPPs to a given test image an N-class classifier is needed. For constructing this classifier from the binary ECs, an aggregation of one-to-one EC classifiers is applied: each EC is trained to distinguish between two IPPs. This is done for all pairs of IPPs, leading to $N(N-1)/2$ binary classifiers each voting for one of the IPPs in its pair. For assigning an IPP to an image under test, these $N(N-1)/2$ classifiers are applied and the final decision is the IPP that receives the majority of the votes.

For training the IPP-classifier we considered only the 11 IPPs discussed in section 4.3.

Figure 6 shows the results of the classification obtained after training and validation on 10000 cover images of each processing pipeline. The figure shows the confusion matrix of the classification. The value in row i, column j is the probability that an image created by the IPP noted on row i is classified as being created by the IPP in column j. For the sake of clarity, zero values are omitted.

The results show that the classifier is capable to identify the different IPPs with a very high accuracy. In particular, it is possible to distinguish between the different types of CFA interpolation. Note also that, except for the LR6Std there is no confusion between the LR6 generated images and those from DC or RT.

Interestingly, when applying the IPP-classifier trained on the cover images for classifying the IPP of the stego images, a very similar classification accuracy Acc [1] is obtained. The obtained Acc is 96.1% and 95.9% for resp. the cover and stego images. This means that, while steganalysis performance is highly dependent on the processing chain, the detection of the processing chain suffers

[1] Acc=ratio of correctly classified items to the total number of classified items; expressed as a percentage

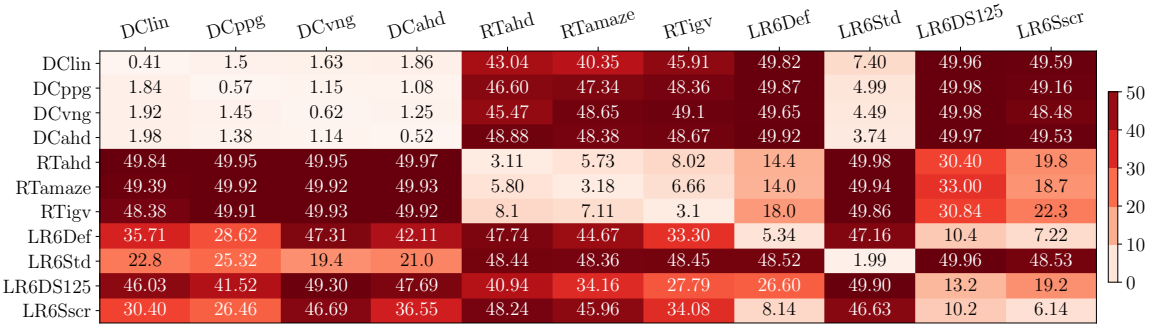

Figure 5: Influence of demosaicing and development software (P_E in %).

only a minor influence of the presence of steganography (for the experiments conducted in this paper).

6 MITIGATION OF THE CSM - THE IPP-INFORMED DETECTOR

The results of the IPP-classifier can be used for selecting the detector that was trained on the closest (and possibly same) source, i.e. an IPP-informed detector. For each tested image, the IPP-classifier discussed in section 5 is applied. In the second step the steganalysis detector that was trained on the detected IPP is applied for deciding whether the test image is cover or stego. This is done for 10000 randomly picked images for each of the considered IPPs.

In this test we also include images developed with DCraw/ AHD/ GC=Prophoto (DCCGPP) and DCraw /AHD/ WB= Camera (DCWBcam). Note that neither of these two processing pipelines were used for training the IPP-classifier or for training the steganalysis detectors.

In the test we also included 10000 JPEG images collected from Flickr. The selected images correspond to images acquired with the Leica M9 camera and with quantization table corresponding to Adobe level 12. The images were center-cropped to a size of 512×512 in a way that preserves the DCT structure. The corresponding stego-images were created with an embedding rate of 0.1 bpnac.

Figure 7 compares steganalysis results obtained by four different detection strategies:

- 'Fully matched' case: training and test images come from the same source. This is the baseline for the comparison. It represents the best results that can be obtained using the chosen feature set and the EC classifier.
- 'Class Boss': training on the classical BossBase, i.e. resized (spatially interpolated) images developed with DCraw using PPG demosaicing as explained in the introduction of the paper,
- 'Mixed training': the results of training the EC on a mix of the 11 sources (consisting of 1000 images from each source) as proposed in [13],
- 'IPP informed': the results of the proposed IPP-informed steganalyzer.

For DCGCPP and DCWBcam the values of the 'Fully matched' case in figure 7 are extracted from figures 3 and 4. For the images downloaded from Flickr the presented values for the 'Fully matched' case are the average results over five 5000/5000 random splits of the available images. For the 'Mixed training' of the Flickr images, we decided to use only images from LR6Def, LR6DS125 and LR6SScr for training (3400 of each IPP).

Figure 7 shows that the 'Class Boss' method clearly suffers the most of the CSM. The 'Mixed training' considerably reduces the impact of the CSM compared to the 'Class Boss' approach, except for the case where the latter is almost fully-matched (DCppg and DCahd). The 'IPP informed' method proposed in the current paper results in the smallest increase of P_E with respect to the fully-matched detector.

The proposed method also behaves better than 'Mixed training' for the two processing pipelines that were not used for training the IPP-classifier or the steganalysis detectors. Note that the increase of P_E w.r.t. the fully matched case is the highest for DCWBcam. However, the increase for both DCWBcam and DCGCPP is much smaller than the mismatch found in respectively figure 3 and 4 between WBdef/WBcam and GCdef/GCprophoto. The type of approach as presented in this paper thus also provides some robustness with respect to unknown IPPs. For the Flickr images the IPP-informed and the mixed training detectors obtain similar results. The large difference w.r.t. the 'Fully matched' case suggests that the LR6 developments used for training our models should be expanded, particularly w.r.t. the down-sampling operations that are present in the Flickr database. We also noted that the IPP-classifier classes 99.7% of the Flickr images as one of the three LR6 developments. The remainder is classified as RT developments.

7 CONCLUSIONS AND FURTHER WORKS

The paper investigates the influence of the image processing pipeline on the cover-source mismatch for JPHide. It also proposes a simple classifier of the IPP and shows how it can be exploited for reducing the CSM due to the IPP. We show that within this setup, the proposed IPP-informed steganalysis outperforms approaches based on mixed training over the examined sources. Partitioning the image data prior to steganalysis thus seems a promising approach for mitigating the CSM (see also [16]).

According to [7], the impact of the image processing pipeline on the CSM is more important than the choice of camera or the image acquisition parameters. The current paper additionally shows that the CSM can be significantly reduced by combining IPP classification with training set design. For the latter the RAW images

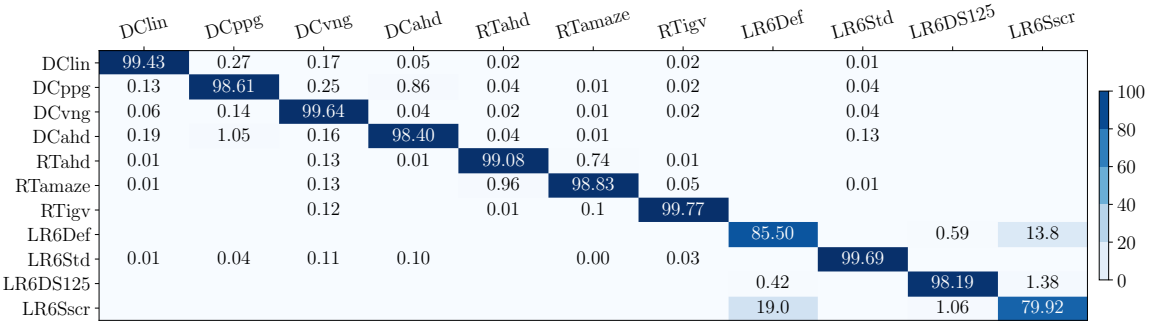

Figure 6: Confusion matrix for the supervised classification of the image processing pipeline (IPP).

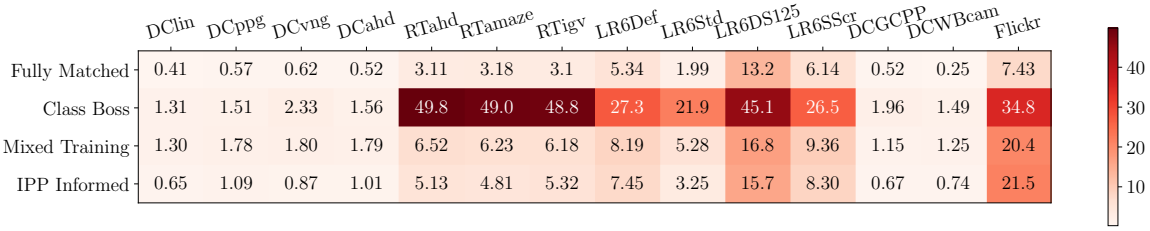

Figure 7: Comparison of P_E (in %) for the four detection strategies.

of BossBase can be used for generating training databases that match or are close to the IPP of the images under investigation. The authors expect that for mitigating the impact of image acquisition a similar approach could be followed, based on a carefully designed expansion of the BossBase, i.e. spanning a larger variation of acquisition parameter settings.

Future work will assess the current methodology on other steganographic schemes in the pixel or JPEG domain.

ACKNOWLEDGMENTS

The authors would like to thank Samuel Tap who, in the frame of an internship at the Royal Military Academy, coded most of the Python™ scripts used for generating the results presented in this paper. This work was partially supported by the French ANR DEFALS program (ANR-16-DEFA-0003) and by the Belgian Royal Higher Institute for Defence (projects DAP16-01 and DAP18-01).

REFERENCES

[1] M. Barni, G. Cancelli, and A. Esposito. 2010. Forensics aided steganalysis of heterogeneous images. In *Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 1690–1693.
[2] P. Bas, T. Filler, and T. Pevny. 2011. Break our Steganographic System - the ins and outs of organizing BOSS. In *13th workshop on Information Hiding*. Springer, Prague, Czech Republic, 59–70.
[3] M. Boroumand and J. Fridrich. 2017. Scalable Processing History Detector for JPEG Images. In *Media Watermarking, Security, and Forensics*. IS&T.
[4] Cambridge in Colours 2017. Photography tutorials: White balance. (2017). Retrieved Jan 24, 2018 from http://www.cambridgeincolour.com/tutorials/white-balance.htm
[5] G. Cancelli, G. Doerr, M. Barni, and I. J. Cox. 2008. A comparative study of ±1 steganalyzers. In *IEEE 10th Workshop on Multimedia Signal Processing*. 791–796.
[6] J. Fridrich, J. Kodovský, V. Holub, and M. Goljan. 2011. Breaking HUGO - The Process Discovery. In *13th Int. Conf. on Information Hiding*. 85–101.
[7] Q. Giboulot, R. Cogranne, and P. Bas. 2018. Steganalysis into the Wild: How to Define a Source?. In *Media Watermarking, Security, and Forensics*. IST.
[8] A.D. Ker, P. Bas, R. Böhme, R. Cogranne, S. Craver, T. Filler, J. Fridrich, and T. Pevnỳ. 2013. Moving steganography and steganalysis from the laboratory into the real world. In *Workshop on Information Hiding and Multi-media Security (IH&MMSEC)*. ACM, 45–58. http://doi.acm.org/10.1145/2482513.2482965
[9] A. Ker and T. Pevny. 2014. A Mishmash of Methods for Mitigating the Model Mismatch Mess. In *Electronic Imaging, Media Watermarking, Security and Forensics*, Vol. SPIE Vol. 9028. SPIE, San Francisco, 90280I.
[10] J. Kodovsky and J. Fridrich. 2012. Steganalysis of JPEG Images Using Rich Models. In *Electronic Imaging, Media Watermarking, Security and Forensics*. SPIE, San Francisco.
[11] J. Kodovsky and J. Fridrich. 2013. Steganalysis in Resized Images. In *Proc. ICASSP*.
[12] J. Kodovsky, J. Fridrich, and V. Holub. 2012. Ensemble classifiers for steganalysis of digital media. *IEEE TIFS* 7, 2 (2012), 432 – 444.
[13] J. Kodovsky, V. Sedighi, and J. Fridrich. 2014. Study of Cover Source Mismatch in Steganalysis and Ways to Mitigate its Impact. In *Electronic Imaging, Media Watermarking, Security and Forensics*. SPIE, SPIE, San Francisco, California, 90280J.
[14] I. Lubenko and A. Ker. 2012. Going from small to large data in steganalysis. In *Media Watermarking, Security, and Forensics*, Vol. 8303. SPIE, 83030M–83030M–10. https://doi.org/10.1117/12.910214
[15] F. Comby M. Yedroudj, M. Chaumont. 2018. How to augment a small learning set for improving the performances of a CNN-based steganalyzer?. In *Media Watermarking, Security, and Forensics*. IS&T.
[16] J. Pasquet, S. Bringay, and M. Chaumont. 2014. Steganalysis with Cover-Source Mismatch and a small learning database. In *Proc. EUSPICO*.
[17] X. Qiu, H. Li, W. Luo, and J. Huang. 2014. A universal Image Forensic Strategy based on Steganalytic Model. In *Workshop on Information Hiding and Multi-media Security (IH&MMSEC)*. Salzburg.
[18] Rawpedia 2015. Gamma differential. (2015). Retrieved Jan 29, 2018 from https://rawpedia.rawtherapee.com/Gamma_-_Differential
[19] M.S. Tooms. 2016. *Colour Reproduction in Electronic Imaging Systems: Photography, Television, Cinematography*. John Wiley & Sons, Chapter Appendix H: Deriving the Standard Formula for Gamma Correction, 667–672.