

Effect of Recompression on Attacking JPEG Steganographic Schemes — An Experimental Study

Yun Q. Shi, Chunhua Chen, Wen Chen, Maala P Kaundinya
New Jersey Institute of Technology, Newark, NJ, USA 07102
{shi,cc86}@njit.edu

Abstract—In the implementation of a few JPEG steganographic schemes such as OutGuess and F5, an additional JPEG compression may take place before data embedding. The effect of this recompression on the performances of steganalyzers is experimentally studied and reported in this paper. Through a group of carefully designed experimental works, we show that the training and testing procedures adopted in classification are of great importance. An improper training and testing procedure may lead to poor steganalysis performance even for a powerful steganalyzer or an accurate performance comparison. Some other informative observations are presented in the paper as well.

I. INTRODUCTION

Steganalysis, the science and art of detecting the very existence of hidden information in a given medium, has attracted many researchers' attentions since the end of last century. Steganalyzers have been categorized into specific and universal two types. Specific steganalysis launches an attack on a targeted steganographic technique, while universal steganalysis aims at all of steganographic schemes and is based on the principle of machine learning. Among the universal schemes, some can detect steganographic methods applied to both the spatial domain and the JPEG domain [e.g., 1,2,3], whereas some are designed for attacking JPEG steganography [e.g., 4,5].

In [1], Farid proposed a universal steganalyzer based on the high order statistical moments of wavelet high-frequency subbands of a given test image, denoted by MW in this paper. It can achieve generally better detection rate than random guess for universal steganalysis purpose.

In [2,3], Xuan et al. and Shi et al. presented another universal steganalysis framework, using the moments of characteristic functions of the given test image, its prediction-error image, and all of their wavelet transform subbands as features, denoted by MC, providing a better performance than [1] in general.

In [4], Fridrich developed a steganalyzer with features obtained by computing some functionals of the given image and its calibrated version, denoted by JF. Designed specifically for JPEG steganography, this scheme performs better than [1,2,3] in attacking JPEG steganographic tools.

In [5], Shi et al. proposed another steganalyzer designed for attacking JPEG steganography, which is based on Markov process and denoted by MP. There, the difference JPEG 2-D

arrays along horizontal, vertical, and diagonal directions are formed, the Markov process is applied, and all of the elements of transition probability matrices are used as features for steganalysis. To greatly reduce computational complexity, a thresholding technique is developed, which is based on a statistical analysis. Experimental works reported in [5] have demonstrated that this scheme outperforms [1,3,4] by a significant margin in attacking modern JPEG steganographic schemes: OutGuess ([6]), F5 ([7]), and MB1 ([8]).

To benchmark steganographic and steganalysis techniques, Kharrazia et al. [9] have compared three steganalysis techniques (including [1,4]) when attacking a few JPEG steganographic techniques (including [6,7,8]). Because some JPEG steganographic tools recompress JPEG images before embedding data and the recompression artifacts could cause false alarm in steganalysis, the so called "confusion tests" were designed to investigate this issue in [9]. There, two types of confusion tests were considered: 1) the original JPEG test image versus the recompressed image with the quality factor estimated from the original image. (this occurs with OutGuess and F5); 2) the original test image versus the recompressed image with a quantization step as twice as the estimated original quantization step (this happens in perturbed quantization (PQ) steganography [10]).

In this paper, performances of the aforesaid steganalyzers in attacking modern JPEG steganography are evaluated. To focus on the effect caused by JPEG recompression (the main theme of this paper), only MB1 is considered as the representative of JPEG steganographic schemes with the considerations presented in Section II. It is expected that the observation obtained in this case study can also be applied to other modern JPEG steganographic schemes.

The rest of this paper is organized as follows. The environment of our experiments is described in Section II. In Sections III, VI, V, and VI, the basic tests, the recompression tests, the cross tests, and the comprehensive tests are presented. Finally, discussions and conclusions are made in Section VII.

II. EXPERIMENT ENVIRONMENT

A. Representative steganographic scheme

Developed by Sallee, model based steganography (MB) [8] is known as an advanced steganographic scheme. Furthermore, the code of MB1, provided in MATLAB and available in public [11], is easy to implement and control. Therefore, the MB1 is selected as the representative JPEG steganography in this investigation. Readers please refer to [8] for more information of MB1.

B. Test image database

In our experimental works, we use all of the 1096 images contained in the CorelDRAW Version 10.0 software CD#3 [12]. These 1096 images, of size either 768×512 or 512×768 , come without JPEG compression, thus facilitating our investigation. Some sample images are shown in Fig. 1. While given in color, the CorelDRAW images are firstly transformed to gray scale images by applying the irreversible color transform (ICT) [13] in our experiments.



Figure 1. Some sample CorelDRAW images used in this work

C. Cover/Stego image generation

We compress these gray scale images, derived from CorelDRAW database, to JPEG images with quality factor $Q=80$ first. The resultant images are our 1st cover-image group (denoted by C_{80}^1). Then we recompress images in C_{80}^1 with $Q=80$ and $Q=60$, respectively, resulting in our 2nd cover-image group (C_{8080}^2) and 3rd cover-image group (C_{8060}^2), respectively. Finally, we use MB1 code to embed randomly generated bits into each image in the three cover-image groups with embedding rate 0.05 bpc (bits per non-zero JPEG AC coefficients). The corresponding stego-image groups are denoted by S_{80}^1 , S_{8080}^2 , and S_{8060}^2 , respectively.

D. Classification

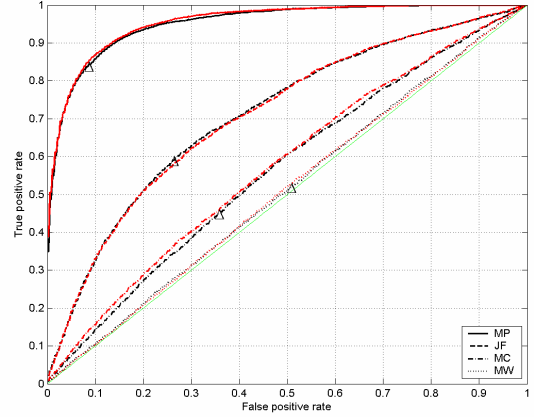
The support vector machine (SVM), a popularly used classifier, is used in this experimental work. The SVM codes in MATLAB are downloaded from [14], which provide four basic kernels: linear, polynomial, radial basis function, and sigmoid. We use the polynomial kernel with degree 2 here.

Both the receiver operating characteristics (ROC) curve ([15]) and the area under the ROC curve (AUC) are used here to describe the performance of a trained classifier. The ROC curves and AUC values reported in this paper are the arithmetic average of 20 random experiments. In each experiment, 913 randomly selected cover images and corresponding 913 stego images are sent to the classifier for

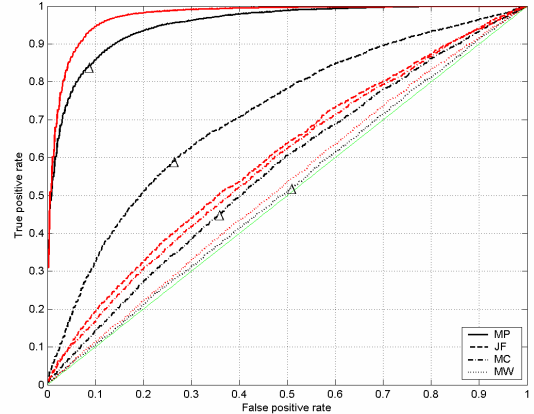
training purpose. The remaining 183 cover images and 183 stego images are then sent to the trained classifier for testing.

III. BASIC TESTS

Basic tests are designed to ensure the difference between a cover image and its corresponding stego image lies only on data embedding instead of different number of JPEG compressions. The experimental results are reported in Figure 2 and Table I. Basic tests include the following three cases: 1. C_{80}^1 vs. S_{80}^1 , 2. C_{8080}^2 vs. S_{8080}^2 , 3. C_{8060}^2 vs. S_{8060}^2 .



(A)



(B)

Figure 2. ROC curves in basic tests. (A). Case 1 (in black and with a “ \triangle ”) and Case 2 (in red). (B). Case 1 (in black and with a “ \triangle ”) and Case 3 (in red).

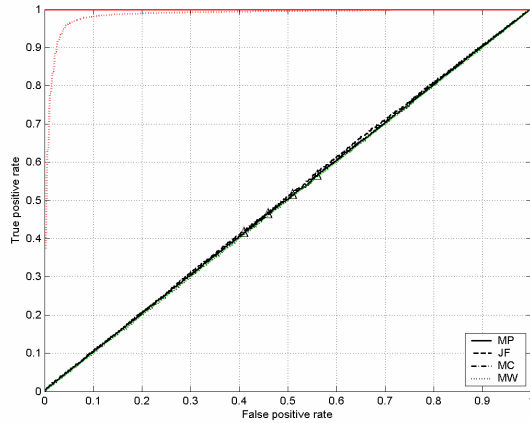
The observations we can draw are as follows: (1) In Case 2, all the steganalyzers give similar performances to that in Case 1, indicating that the change caused by an additional JPEG compression with the same Q -factor is almost negligible. This agrees with our other extensive experimental works. (2) In Case 3, the performances of MW, MC, and MP are slightly improved, while the performance of JF becomes

worse, when compared to that in Case 1. (3) In all the cases, MP gives the best performance.

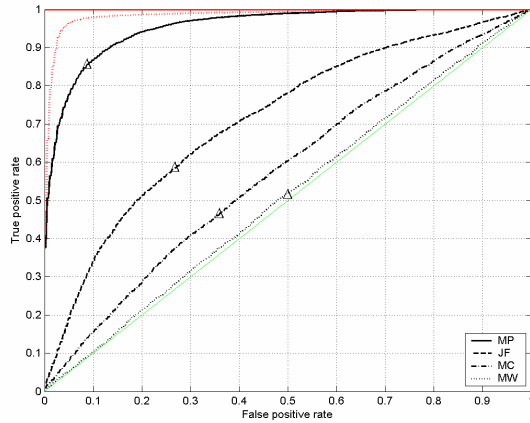
IV. RECOMPRESSION TESTS

Recompression tests, also referred to as one-additional-compression tests, contain the following four cases: 4. C_{80}^1 vs. C_{8080}^2 , 5. C_{80}^1 vs. C_{8060}^2 , 6. C_{80}^1 vs. S_{8080}^2 , 7. C_{80}^1 vs. S_{8060}^2 .

Case 4 and Case 5 focus on these steganalysis tools' capability in classifying a cover image and its recompressed version. Case 6 and Case 7 test the performance of these steganalysis tools on distinguishing a cover image and its stego image obtained from its recompressed version.



(A)



(B)

Figure 3. ROC curves in recompression tests. (A). Case 4 (in black and with a “ Δ ”) and Case 5 (in red; the curves of MC, JF, and MP are merged to the upper and left boundaries of the graph). (B). Case 6 (in black and with a “ Δ ”) and Case 7 (in red; the curves of MC, JF, and MP are merged to the upper and left boundaries of the graph).

From the test results shown in Fig. 3 and Table I, we have the following observations: (1) In Case 4, all of these four steganalyzers cannot distinguish a cover image and its

recompressed image with the same Q-factor because, as mentioned in Section III, the changes caused by recompression with the same Q-factor are trivial. (2) In Case 5, all four steganalyzers can easily distinguish a cover image and its recompressed version with a Q-factor that provides a quantization step as twice large as that used in the first compression. (3) In Case 6, the performances of these steganalyzers are very close to that in Case 1. (4) The performances of these steganalyzers in Case 7 are very close to that in Case 5. This indicates that the recompression with Q-factor 80 followed by Q-factor 60 causes severe statistical artifacts that make the effect of statistical artifacts caused by MB1 data embedding submerged.

V. CROSS TESTS

Two cases are studied in cross tests: Case 8. In training: C_{80}^1 vs. S_{8080}^2 , In testing: C_{8080}^2 vs. S_{80}^1 , Case 9. In training: C_{80}^1 vs. S_{8060}^2 , In testing: C_{8060}^2 vs. S_{80}^1 .

Cross tests are designed to test the steganalyzers' performances when they are insufficiently trained. The test results are shown in Figure 4 and Table I.

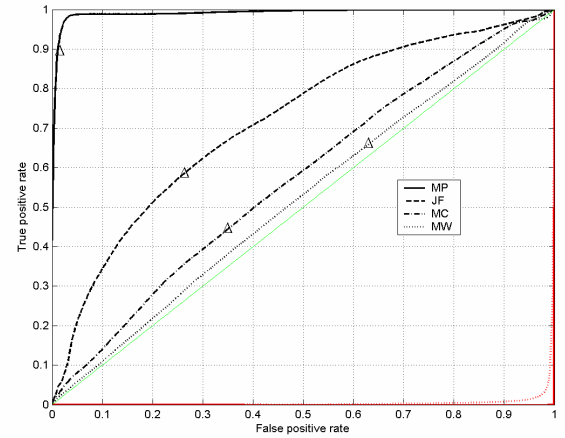


Figure 4. ROC curves in cross tests: Case 8 (in black and with a “ Δ ”) and Case 9 (in red; the curves of MC, JF, and MP are merged to the right and bottom boundaries of the graph).

Observations are as follows: (1) In Case 8, the detection rates of each steganalyzer are enhanced (with that of MP enhanced the most as compared to Case 1). (2) In Case 9, all the steganalysis tools fail, i.e., a cover image is recognized as a stego image and vice versa. This can be explained using our previous comments, i.e., JPEG compression with Q-factor 80 followed by Q-factor 60 causes severe statistical artifacts, which is stronger than that caused by data hiding.

VI. COMPREHENSIVE TESTS

To avoid effects caused by recompression and insufficient training in steganalysis, we designed comprehensive tests as follows: Cases 10. (C_{80}^1 and C_{8080}^2) vs. (S_{80}^1 and S_{8080}^2), 11. (C_{80}^1 and C_{8060}^2) vs. (S_{80}^1 and S_{8060}^2).

The test results are shown in Figure 5 and Table I.

It is seen that: (1) In Case 10, the detection rates of each steganalyzer are slightly improved. (2) In Case 11, MC and MW give similar performance to that of Case 1, and MP gives slightly enhanced performance, while JF's performance worsens.

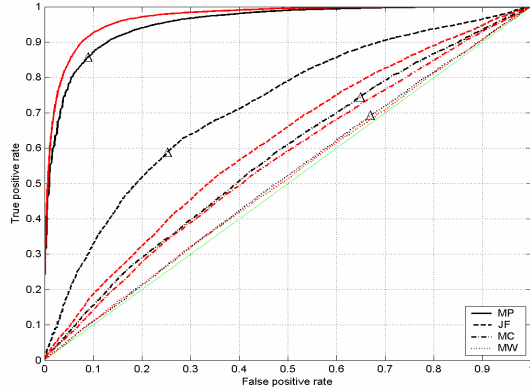


Figure 5. ROC curves in comprehensive tests: Case 10 (in black and with a “ \triangle ”) and Case 11 (in red).

TABLE I. AUC (AREA UNDER ROC CURVE) IN OUR EXPERIMENTS

	MW [1]	MC [3]	JF [4]	MP [5]
Case 1	0.50651	0.56118	0.70584	0.94568
Case 2	0.50595	0.57034	0.70406	0.94879
Case 3	0.52103	0.58300	0.59402	0.96786
Case 4	0.49517	0.49839	0.50257	0.49887
Case 5	0.97946	0.99453	0.99454	0.99454
Case 6	0.50742	0.56998	0.70697	0.94866
Case 7	0.97876	0.99453	0.99454	0.99454
Case 8	0.52102	0.56982	0.71568	0.99053
Case 9	0.00564	0.00000	0.00000	0.00000
Case 10	0.51303	0.57385	0.71541	0.95069
Case 11	0.51022	0.56089	0.60886	0.96655

VII. DISCUSSION AND CONCLUSION

1) The recompression tests (Cases 4 and 5), which do not involve any stego image in training and testing, cannot shed light to steganalyzers' performance. This is because the performances of all four steganalyzers are very similar in these two designed cases.

2) Cross tests (Cases 8 and 9) have insufficient training and therefore should not be adopted in steganalysis to handle recompression issue.

3) Basic tests (Cases 1,2,3) tell us that there should have been no additional compression in steganalysis performance evaluation. That is, the cover image and stego image should not differ by one or multiple JPEG compression in order to achieve reliable steganalysis and to achieve objective performance comparison among steganalysers.

4) As said, the OutGuess and F5 softwares do involve an additional compression in data embedding. And, in reality,

given JPEG test images may have gone through different number of JPEG compressions. Hence, it is necessary to design proper training and testing procedure to handle recompression case.

5) Comprehensive tests, involving both images with and without an additional compression in training, appear to be a promising way to handle the recompression issue.

6) It is noticed that the performance of JF declines given recompression with a smaller Q-factor (refer to Cases 3 and 11 as compared with Case 1). The reason behind this phenomenon is that the calibration operation, the key step of JF, does not work well for this kind of recompression.

7) Repeating JPEG compression with the same Q-factor causes negligible changes to images.

8) JPEG compression with Q-factor 80 followed by Q-factor 60, i.e., the quantization step used in the 2nd compression is almost doubled as that in the 1st compression, the statistical artifacts are more severe than that caused by MBI data embedding with the embedding rate of 0.05 bpc (bits per non-zero AC coefficient). Our additional tests, indicate that this is true for 0.1 bpc and 0.2 bpc as well. This observation is also valid for JPEG compression with Q-factor 80 followed by Q-factor 90.

REFERENCES

- [1] H. Farid, “Detecting hidden messages using higher-order statistical models”, *International Conference on Image Processing*, Rochester, NY, USA, 2002.
- [2] G. Xuan, Y. Q. Shi, J. Gao, D. Zou, C. Yang, Z. Zhang, P. Chai, C. Chen, and W. Chen, “Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions”, *Information Hiding Workshop 2005*, Barcelona, Spain, June 2005.
- [3] Y. Q. Shi, G. Xuan, D. Zou, J. Gao, C. Yang, Z. Zhang, P. Chai, W. Chen, and C. Chen, “Steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network”, *International Conference on Multimedia and Expo*, Amsterdam, Netherlands, 2005.
- [4] J. Fridrich, “Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes”, *Information Hiding Workshop 2004*, Toronto, ON, Canada, 2004.
- [5] Y. Q. Shi, C. Chen, and W. Chen, “A Markov process based approach to effective attacking JPEG steganography”, *Information Hiding Workshop 2006*, Old Town Alexandria, VA, USA, July 10-12, 2006.
- [6] N. Provos, “Defending against statistical steganalysis”, *10th USENIX Security Symposium*, Washington DC, USA, 2001.
- [7] A. Westfeld, “F5 a steganographic algorithm: high capacity despite better steganalysis”, *Information Hiding Workshop 2001*, Pittsburgh, PA, USA, 2001.
- [8] P. Sallee, “Model-based methods for steganography and steganalysis”, *International Journal of Image and Graphics*, 5(1): 167-190, 2005.
- [9] M. Kharrazi, H. T. Sencar, and N. D. Memon, “Benchmarking steganographic and steganalysis techniques”, *Security, Steganography, and Watermarking of Multimedia Contents 2005*, San Jose, CA, USA, 2005.
- [10] J. Fridrich, M. Goljan, and D. Soukal, “Perturbed quantization steganography with wet paper codes”, *ACM Multimedia Workshop*, Magdeburg, Germany, September 20-21, 2004.
- [11] [Online] <http://redwood.ucdavis.edu/phil/papers/iwdw03.htm>.

- [12] <http://www.corel.com/>.
- [13] M. Rabbani and R. Joshi, "An overview of the JPEG2000 still image compression standard", *Signal Processing: Image Communication*, 2002, 17(1): 3-48.
- [14] C. C. Chang and C. J. Lin: LIBSVM: a library for support vector machines, 2001. [Online] <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [15] T. Fawcett, "Roc graphs: notes and practical considerations for researchers". [online] http://home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf.