https://doi.org/10.2352/ISSN.2470-1173.2018.07.MWSF-160

Deep Learning Regressors for Quantitative Steganalysis

Mo Chen, Mehdi Boroumand, and Jessica Fridrich, Department of ECE, SUNY Binghamton, NY, USA, mochen8@gmail.com, {mboroum1, fridrich}@binghamton.edu

Abstract

The goal of quantitative steganalysis is to provide an estimate of the size of the embedded message once an image has been detected as containing secret data. For steganographic algorithms free of serious design flaws, such as schemes based on least significant bit replacement, the most competitive quantitative detectors have traditionally been built as regressors in rich media models. Considering the recent advances in binary steganalysis due to deep learning, in this paper we use the features extracted from the activation of such CNN detectors for the task of payload estimation. The merit of the proposed architecture is demonstrated experimentally on steganographic algorithms operating both in the spatial and JPEG domain.

Introduction

Steganography is the art of communicating secret messages to another party by hiding the secrets in cover objects so that an adversary monitoring the traffic cannot distinguish between genuine cover objects and objects carrying secret data. Formally, steganography is considered broken when the mere presence of the secret can be established. Forensic analysts, however, are likely to benefit from accessing additional information, such as what algorithm was used to hide the secret and how long the message is. While steganalysis can be formulated as a binary hypothesis test, determining the payload size is an estimation problem. Although the output of a binary classifier could be mapped to an approximate payload size, such estimators are rarely the best. Conversely, the output of a quantitative steganalyzer is not necessarily the best test statistic [23].

The objective of quantitative steganalysis is to estimate the number of embedding changes, which can be related to the message length after taking into consideration the source coding applied during embedding [10]. Historically, the first accurate detectors of Least Significant Bit (LSB) replacement were quantitative detectors, such as RS analysis [12], Sample Pairs analysis (SPA) [9], Triples analysis [18], and the Weighted-Stego (WS) detector [11, 19, 6, 33]. These so-called *structural* attacks are fundamentally possible because of the fixed polarity of changes imposed by LSB replacement. In this case, detection of stego signal applied to all pixels amounts to detecting a known deterministic signal, which facilitates construction of very accurate detectors and payload estimators. This is because flipping the LSB changes the pixel mean while the embedding operation of LSB matching (also known as ± 1 embedding) changes the variance while preserving the pixel mean, which makes it much harder to detect. Structural quantitative detectors are thus fundamentally limited and do not generalize to embedding based on LSB matching.

An alternative and general approach to quantitative steganalysis was proposed in [24] by formulating the problem of message-length estimation as a regression in a suitably chosen representation of images (feature space). A quantitative steganalyzer constructed in this way can be built for an arbitrary embedding method, and its performance generally depends on how sensitive the features are to embedding and how detectable the embedding is using binary classifiers. The price for such flexibility is the need for a training phase in which the regressor is presented with samples of features extracted from a database of stego images embedded with a range of payloads. The same paper showed the benefit of using non-linear regressors, which were implemented using support vector regression. The complexity of training such regressors limited the dimensionality of the feature space one could use to build the payload size estimator. To permit the utilization of more complex and high-dimensional image descriptors called rich media models [13, 20, 4, 28, 8, 16, 27, 7], the authors of [21] proposed a variant of a regression tree modified to reflect the specifics of steganalysis and approximate the regression function by a generalized additive model while improving the quality of the fit sequentially in a gradient-descent manner.

Recently, novel steganalysis detector architectures implemented within the paradigm of deep Convolutional Neural Networks (CNN) have been proposed. The first architecture with respectable performance employed Gaussian activation function and a high-pass preprocessing layer [25, 26]. The architecture proposed by Xu et al. [31, 30] (XuNet) designed for steganalysis of spatial domain embedding algorithms achieved performance comparable to classical steganalysis with rich media models and the ensemble classifier [22]. A markedly better detection of spatial-domain steganography was recently achieved with an eight-layer network called the YeNet [32], which constitutes the current state of the art to the best knowledge of the authors (as of December 2017). For JPEG steganalysis, two recently proposed architectures showed a performance improvement over steganalysis with selection-channel-aware Gabor Filter Residuals (GFR) [7]: XuNet made aware of JPEG phase [5] and the deep architecture with shortcut connections [29] proposed to detect J-UNIWARD [17].

In this paper, we adapt deep learning for quantitative steganalysis in both spatial and JPEG domains. Our design, which we call the "bucket estimator," starts by first training a family of CNN detectors, each for a different

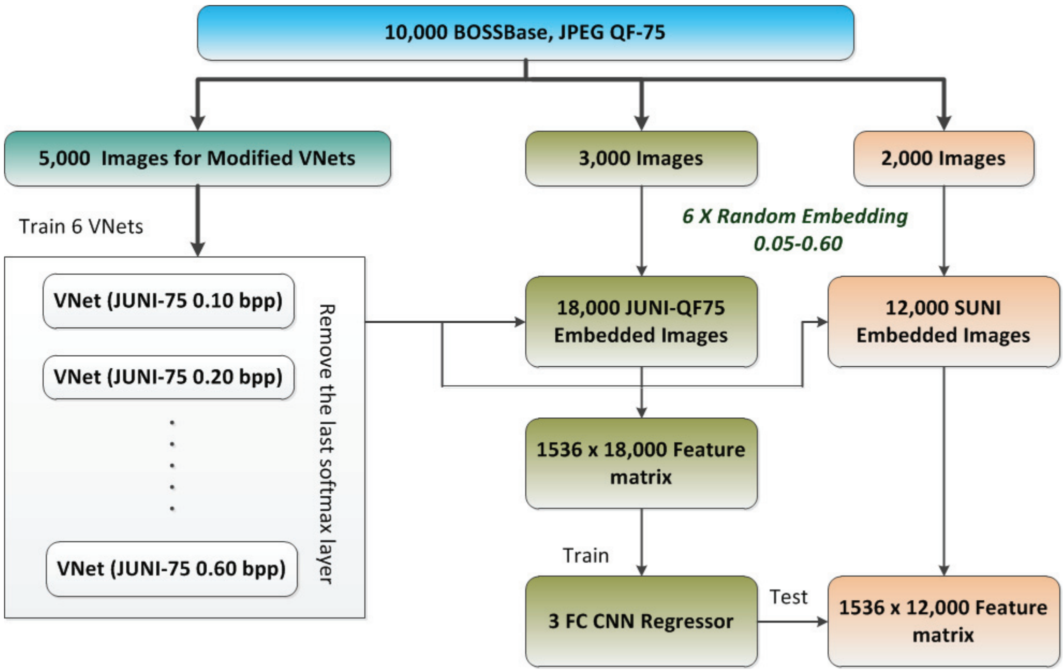

Figure 1. Example of dataset preparation and training for J-UNIWARD with quality factor 75.

fixed payload, and then using their concatenated feature maps as a feature on which a fully-connected network (regressor) is trained by using the Mean Square Error (MSE) as the loss function. This design came out as the best performer among other natural choices. Experiments with two steganographic algorithms in each domain are used to show the merit of the proposed idea.

Bucket estimator

The most natural way to convert a binary classifier built as a CNN into a quantitative regressor is to replace the *softmax* loss function with the MSE and use the embedded payloads as continuous-valued class labels. We have experimented with different approaches, such as initializing the net weights with a pre-trained binary classifier, including multiple stego images with different payloads into the same mini-batch, adopting other loss functions, including the relative estimation error, and expanding the fully connected part of the regressor with different non-linear activation functions. Even though we observed some improvement over the state of the art, the regression trees on rich models [21], we were unable to match the performance of the bucket estimator described next.

The approach that showed the most promise is based on first constructing a bucket of k binary CNN detectors D_{α_i} trained on the cover class and the class of stego images embedded with a fixed payload α_i bpp, $i = 1, \ldots, k$. The feature extraction part of these detectors (the last M activation features connected to the classifier part, the fully connected layers) were then concatenated into a $k \times M$ dimensional feature vector and a payload regressor shown in Figure 2 was trained on such "bucket features" of stego images embedded with payloads α chosen uniformly ran-

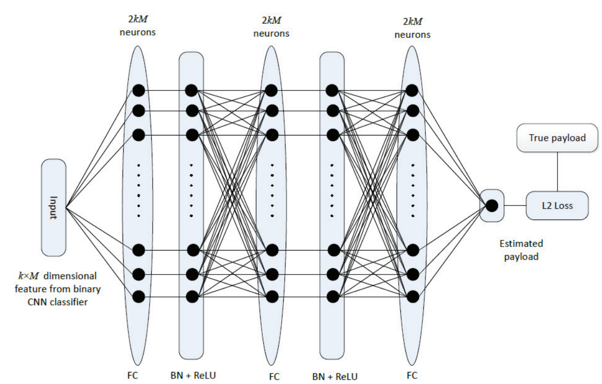

Figure 2. Three-layer FNN payload regressor used in both stego domains.

α	WOW P_{E}	S-UNI P_{E}
0.1	0.2796	0.3452
0.2	0.2092	0.2626
0.3	0.1428	0.1861
0.4	0.1107	0.1324
0.5	0.0820	0.0997
0.6	0.0692	0.0764

Table 1. Detection error P_{E} of individual binary detectors D_{α_i}, $i = 1, \ldots, 6$, trained for a range of payloads α_i for WOW and S-UNIWARD.

Features used	WOW MSE	WOW MAE	S-UNIWARD MSE	S-UNIWARD MAE
0.1	0.0157	0.1006	0.0156	0.0983
0.2	0.0143	0.0954	0.0135	0.0896
0.3	0.0126	0.0882	0.0124	0.0863
0.4	0.0127	0.0889	0.0121	0.0850
0.5	0.0121	0.0857	0.0123	0.0858
0.6	0.0125	0.0871	0.0121	0.0847
All	0.0112	0.0816	0.0109	0.0789

Table 2. Performance of FNN regressors when using the feature maps from one or six payload CNN detectors D_{α_i} for WOW and S-UNIWARD.

Embedding	Bucket+FNN MSE	Bucket+FNN MAE	Bucket+RT MSE	Bucket+RT MAE	SRM+RC+RT MSE	SRM+RC+RT MAE
WOW	.0109	.0789	.0104	.0777	.0151	.0966
SUNI	.0112	.0816	.0109	.0808	.0145	.0922

Table 3. MSE and MAD of three different regressors for spatial domain steganography: the bucket regressor, regression tree on bucket regressor features, and regression tree on SRM features transformed with random conditioning.

domly from some fixed interval $\mathcal{I}$. The regressor is a three-layer fully connected neural network (FNN) with$2kM$ neurons in each layer and an output neuron. This regressor uses batch normalization and the ReLU non-linearity in all non-output layers.

For spatial domain steganography, the detectors D_{α_i} were implemented as YeNets without the knowledge of the selection channel (TLU CNN in the original publication [32]) because the payload is the unknown parameter to be estimated. The dimensionality of the feature vector – the concatenated feature maps before the classifier in a YeNet – is $M = 16\times 3\times 3 = 144$. Since we selected a bucket of $k = 6$ detectors trained for $\alpha_i \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ bpp, the resulting feature representation of images had dimensionality $k\times M = 6\times 144 = 864$. The image source was the BOSSbase 1.01 database [1] downsampled to 256×256 using default 'imresize' in Matlab. A random half of the images (5000 cover and stego images) were used for training the detectors D_α, where 4,000 pairs were used for training and 1,000 pairs for validation. As in [32], downsampled images from BOWS2 [2] (all 10,000 of them) were added to the training set to prevent the YeNet from overfitting. Out of the remaining 5,000 BOSSbase cover-stego pairs, 3,000 of them were used to train the regressor and 2,000 were used to assess the regressor's accuracy. The 5,000 stego images were each embedded with six payloads randomly chosen from the interval $\mathcal{I} = [0.05, 0.6]$, making the total number of training and testing images $3,000\times 6 = 18,000$ and $12,000$, respectively. The regressor was a three-layer fully connected network with $2\times 864 = 1,728$ neurons in each layer and an output neuron.

For JPEG steganography, we used the VNet [5] with a bucket of $k = 6$ detectors for payloads $\alpha_i \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ bpnzac (bits per non-zero AC DCT coefficient). The VNet was modified in comparison to the original publication [5] in the following manner. To reduce the feature sparsity and decrease the feature dimensionality of the bucket, the number of features was reduced from 512 to 256. The resulting feature dimensionality for training the regressor was thus $6\times 256 = 1536$. Similarly, the regressor was a three-layer fully connected network with 3,072 neurons in each layer and an output neuron. The image source was the BOSSbase 1.01 (the original 512×512 images) compressed with quality factors 75 and 95 because we could afford to train the VNet for the original non-resized BOSSbase images. Since the VNet is smaller, it does not benefit from adding BOWS2 images as much as YeNet, which is why we only trained on BOSSbase images in contrast to the spatial domain. As in the spatial domain, a random half of BOSSBase images were used for training each binary classifier D_α with 4,000 pairs for training and 1,000 pairs for validation. and the other 3,000 and 2,000 were used for training and assessing the regressor. The training and testing libraries for the regressor were also constructed in a similar way as in the spatial domain. An example of the dataset preparation and the training of the binary classifiers and the regressor is shown schematically in Figure 1 for J-UNIWARD, quality factor 75.

The YeNet and VNet were trained with data augmentation (random rotation and mirroring applied to images). The training hyperparameters were kept the same as in the corresponding publications [32, 5] with one exception. In order to maximize the feature diversity, when trainining the binary CNN detector for each payload, the networks were initialized with different random seeds and all trained from scratch (curriculum training as in [5] and [32] was not used). Additionally, the training and validation sets have been split with different random seeds as well.

For the regressor, a simple minibatch stochastic gradient descend was used. The momentum and weight decay were fixed to 0.9 and 0.01, respectively. The learning rate for all parameters was chosen logarithmically spaced between 10^{-4} and 10^{-6} for 100 epochs. The minibatches were formed by 150 images with different payloads originating from 25 cover images. In other words, each minibatch contained 25 subsets of *6 features corresponding to six stego images, each with a different payload*. The convolution kernels were initialized with a zero-mean Gaussian distribution with standard deviation 0.01 and all biases were disabled.

Experiments

This section reports the results of all experiments. Two steganographic algorithms were tested in each embedding domain. In addition, two JPEG quality factors were used for JPEG images. Two measures of statistical spread were used to compare the bucket regressor with regression trees with rich models – the MSE and the Mean Absolute Error (MAE). We note that a trivial estimator that always outputs the mean payload from the considered range $\mathcal{I} = [0.05, 0.6]$ has $\mathrm{MSE} = (0.6-0.05)^2/12 = 0.252$ and $\mathrm{MAE} = (0.6-0.05)/4 = 0.138$, respectively.

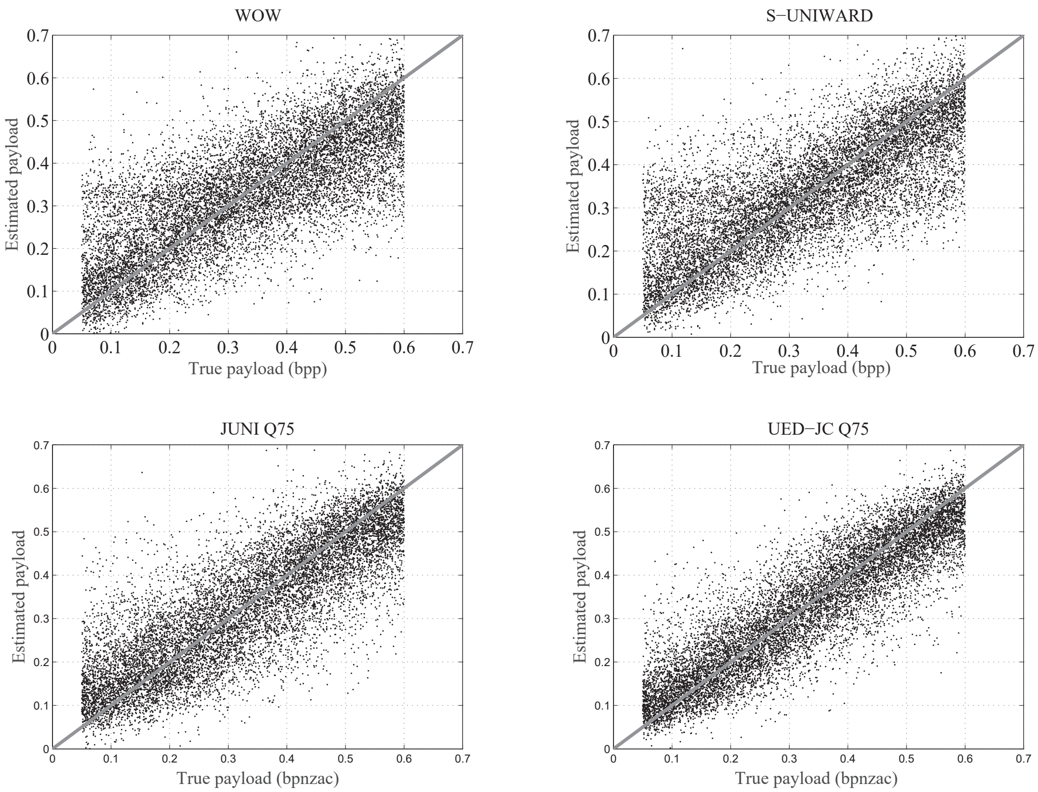

Figure 3. *True vs. estimated payload for the bucket regressor. Upper left WOW, right S-UNIWARD. Bottom left: J-UNIWARD, right UED, both quality factor 75.*

Spatial domain

In the spatial domain, two content-adaptive embedding algorithms were selected for the experiments – WOW [15] and S-UNIWARD [17].

Table 1 shows the performance of six individual detectors D_α in terms of the minimal total probability error under equal priors[1]

$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2}(P_{\mathrm{FA}} + P_{\mathrm{MD}}) \tag{1}$$

for both embedding algorithms. The scatter plot of the bucket payload regressor utilizing the feature maps from all six detectors is shown in Figure 3 top. Table 2 shows the gain of the bucket regressor compared to the regressors built from features of a single binary classifier D_{α_i}. It shows that using the bucket of features helps decrease the estimation error.

In Table 3, we provide the comparison between the bucket regressor and previous art. The first column shows the performance of the bucket regressor as described in this text, the second shows the errors of the regression trees [21] built with features extracted from all individual CNN detectors, while the third column contains the performance of the regression tree trained on SRM features normalized with random conditioning (RC) [3]. The bucket regressor enjoys about 30% smaller MSE than regression trees with randomly conditioned SRM. The FNN regressor performs approximately the same as the regression tree (the first versus the second column). We selected the SRM because the selection-chanel-aware maxSRM [8] cannot be applied because the payload is not known.

JPEG domain

For JPEG domain, J-UNIWARD [17] and UED-JC [14] were tested at JPEG quality 75 and 95.

Table 4 shows the performance of individual detectors D_{α_i} in terms of P_{E} for for both algorithms and quality factors. The scatter plot of the bucket payload regressor utilizing the feature maps from all six detectors is shown in Figure 3 bottom. Table 5 shows the gain in terms of MSE and MAE of using the bucket features versus the regressor trained only on features from a single binary classifier D_{α_i}.

In Table 6, we compare the performance of the bucket regressor (the first column), the regression tree on features used by the bucket regressor (the second column), and the regression tree implemented with GFR features [27]. Again, since the payload is to be estimated, it was not possible to use the selection-channel-aware GFR features [7].

Similar to the spatial domain, the bucket regressor provides about 30% smaller MSE than regression trees trained with the GFR model. As shown in Tables 6 and 3, the FNN regressor on bucket feature maps has a similar performance as a regression tree on the same features.

[1] P_{FA} and P_{MD} are the false-alarm and missed-detection rates.

α	JUNI 75 P_{E}	JUNI 95 P_{E}	UED 75 P_{E}	UED 95 P_{E}
0.1	0.4040	0.4725	0.2450	0.4340
0.2	0.2480	0.4285	0.1070	0.3095
0.3	0.1430	0.3485	0.0550	0.2180
0.4	0.0795	0.2960	0.0330	0.1480
0.5	0.0460	0.2125	0.0150	0.0850
0.6	0.0240	0.1310	0.0080	0.0455

Table 4. Detection error P_{E} of individual detectors trained for a range of payloads for J-UNIWARD and UED-JC.

	JUNI 75		JUNI 95		UED 75		UED 95	
Features used	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
0.1	0.0124	0.0872	0.0196	0.1145	0.0077	0.0672	0.0201	0.1152
0.2	0.0096	0.0757	0.0196	0.1148	0.0059	0.0583	0.0154	0.0986
0.3	0.0090	0.0732	0.0183	0.1089	0.0060	0.0592	0.0139	0.0926
0.4	0.0085	0.0712	0.0181	0.1096	0.0063	0.0609	0.0125	0.0867
0.5	0.0088	0.0726	0.0174	0.1068	0.0062	0.0608	0.0120	0.0855
0.6	0.0090	0.0731	0.0175	0.1071	0.0064	0.0612	0.0116	0.0840
All	0.0082	0.0689	0.0165	0.1032	0.0052	0.0549	0.0107	0.0799

Table 5. Performance of CNN regressors when using the feature mapes from one or six payload detectors D_{α_i} for J-UNIWARD and UED-JC and quality factors 75 and 95.

	Bucket+FNN		Bucket+RT		GFR+RT	
Embedding	MSE	MAE	MSE	MAE	MSE	MAE
JUNI 75	.0082	.0689	.0080	.0694	.0126	.0883
JUNI 95	.0165	.1032	.0160	.1026	.0251	.1247
UED-JC 75	.0052	.0549	.0053	.0556	.0072	.0659
UED-JC 95	.0107	.0799	.0100	.0779	.0165	.1011

Table 6. MSE and MAD of three different regressors for two JPEG embedding algorithms and two quality factors: the bucket regressor, regression tree on bucket regressor features, and regression tree on GFR features.

Conclusions

Quantitative steganalysis deals with the problem of estimating the length of the secret message. In this paper, we propose a new approach to building quantitative steganalyzers (payload estimators) by leveraging the recent progress in binary steganalysis using deep CNNs. A family of such binary classifiers is constructed for a range of fixed payload sizes. The feature maps outputted by such network detectors right before the fully-connected classifier part of the network are concatenated and used as an input into a non-linear regressor implemented with a three-layer fully connected network. This "bucket" estimator provides about 30% reduction in the mean square error of the payload estimator when compared with previous art – regression trees on rich media models. This level of improvement was observed in both the spatial and JPEG domain.

In general, the accuracy of the bucket regressor is strongly related to the performance of the binary detectors. It is to be expected that further advancements in binary steganalysis will lead to corresponding improvements of payload regressors.

All code used to produce the results in this paper, including the network configuration files are available from `http://dde.binghamton.edu/download/`.

Acknowledgments

The work on this paper was supported by the Air Force Office of Scientific Research under the research grant 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. Springer Berlin Heidelberg.

[2] P. Bas and T. Furon. BOWS-2. `http://bows2.ec-lille.fr`, July 2007.

[3] M. Boroumand and J. Fridrich. Non-linear feature normalization for steganalysis. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017.

[4] L. Chen, Y.Q. Shi, P. Sutthiwan, and X. Niu. A novel mapping scheme for steganalysis. In Y.Q. Shi, H.-J. Kim, and F. Perez-Gonzalez, editors, *International Workshop on Digital Forensics and Watermaking*, volume 7809 of *LNCS*, pages 19–33. Springer Berlin Heidelberg, 2013.

[5] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich. JPEG-phase-aware convolutional neural network for steganalysis of JPEG images. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Se-*

curity, Philadelphia, PA, June 20–22, 2017.

[6] R. Cogranne, C. Zitzmann, L. Fillatre, F. Retraint, I. Nikiforov, and P. Cornu. A cover image model for reliable steganalysis. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 178–192, Prague, Czech Republic, May 18–20, 2011.

[7] T. Denemark, M. Boroumand, and J. Fridrich. Steganalysis features for content-adaptive JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 11(8):1736–1746, August 2016.

[8] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich. Selection-channel-aware rich model for steganalysis of digital images. In *IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, December 3–5, 2014.

[9] S. Dumitrescu, X. Wu, and Z. Wang. Detection of LSB steganography via Sample Pairs Analysis. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of Lecture Notes in Computer Science, pages 355–372, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[10] T. Filler, J. Judas, and J. Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security*, 6(3):920–935, September 2011.

[11] J. Fridrich and M. Goljan. On estimation of secret message length in LSB steganography in spatial domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 23–34, San Jose, CA, January 19–22, 2004.

[12] J. Fridrich, M. Goljan, and R. Du. Detecting LSB steganography in color and gray-scale images. *IEEE Multimedia, Special Issue on Security*, 8(4):22–28, October–December 2001.

[13] 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.

[14] L. Guo, J. Ni, and Y. Q. Shi. Uniform embedding for efficient JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 9(5):814–825, May 2014.

[15] V. Holub and J. Fridrich. Designing steganographic distortion using directional filters. In *Fourth IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5, 2012.

[16] V. Holub and J. Fridrich. Phase-aware projection model for steganalysis of JPEG images. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, pages 0T 1–11, San Francisco, CA, February 8–12, 2015.

[17] V. Holub, J. Fridrich, and T. Denemark. Universal distortion design for steganography in an arbitrary domain. *EURASIP Journal on Information Security, Special Issue on Revised Selected Papers of the 1st ACM IH and MMS Workshop*, 2014:1, 2014.

[18] A. D. Ker. A general framework for structural analysis of LSB replacement. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Information Hiding, 7th International Workshop*, volume 3727 of Lecture Notes in Computer Science, pages 296–311, Barcelona, Spain, June 6–8, 2005. Springer-Verlag, Berlin.

[19] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In E. J. Delp, P. W. Wong, J. Dittmann, and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 5 1–17, San Jose, CA, January 27–31, 2008.

[20] 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.

[21] J. Kodovský and J. Fridrich. Quantitative steganalysis using rich models. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2013*, volume 8665, pages 0O 1–11, San Francisco, CA, February 5–7, 2013.

[22] 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.

[23] T. Pevný. Detecting messages of unknown length. In A. Alattar, N. D. Memon, E. J. Delp, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security and Forensics III*, volume 7880, pages OT 1–12, San Francisco, CA, January 23–26, 2011.

[24] T. Pevný, J. Fridrich, and A. D. Ker. From blind to quantitative steganalysis. *IEEE Transactions on Information Forensics and Security*, 7(2):445–454, 2011.

[25] Y. Qian, J. Dong, W. Wang, and T. Tan. Deep learning for steganalysis via convolutional neural networks. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, pages 0J 1–10, San Francisco, CA, February 8–12, 2015.

[26] Y. Qian, J. Dong, W. Wang, and T. Tan. Learning and transferring representations for image steganalysis using convolutional neural network. In *IEEE International Conference on Image Processing (ICIP)*, pages 2752–2756, September 25–28, 2016.

[27] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang. Steganalysis of adaptive JPEG steganography using 2D Gabor filters. In A. Alattar, J. Fridrich, N. Smith, and P. Comesana Alfaro, editors, *The 3rd ACM Workshop on Information Hiding and Multimedia Security*,

IH&MMSec '15, Portland, OR, June 17–19, 2015.

[28] W. Tang, H. Li, W. Luo, and J. Huang. Adaptive steganalysis against WOW embedding algorithm. In S. Katzenbeisser, R. Kwitt, and A. Piva, editors, *The 2nd ACM Workshop on Information Hiding and Multimedia Security*, pages 91–96, Salzburg, Austria, June 11–13, 2014.

[29] G. Xu. Deep convolutional neural network to detect J-UNIWARD. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017.

[30] G. Xu, H.-Z. Wu, and Y. Q. Shi. Ensemble of CNNs for steganalysis: An empirical study. In F. Perez-Gonzales, F. Cayre, and P. Bas, editors, *The 4th ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '16, pages 5–10, Vigo, Spain, June 20–22, 2016.

[31] G. Xu, H. Z. Wu, and Y. Q. Shi. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712, May 2016.

[32] J. Ye, J. Ni, and Y. Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, November 2017.

[33] C. Zitzmann, R. Cogranne, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu. Statistical decision methods in hidden information detection. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 163–177, Prague, Czech Republic, May 18–20, 2011.

Author Biography

Mo Chen received the BS and MS degrees in Electrical Engineering from Shandong University, China, in 1998 and 2001 and the Ph.D. in Electrical Engineering from Binghamton University, State University of New York, in 2006. From 2006 to 2017, he worked as a postdoc and adjunct research scientist at Binghamton University. Since 2007, he has been working as a chief machine vision engineer at JADAK LLC, NY (Novanta) responsible for developing machine vision OEM systems for healthcare automation and clinical analysis applications. His research interests include machine vision and machine learning, digital image and video processing, and digital forensics.

Mehdi Boroumand received his B.S. degree in electrical engineering from the K. N. Toosi University of Technology, Iran, in 2004 and his M.S. degree in electrical engineering from the Sahand University of Technology, Iran in 2007. He is currently pursuing his Ph.D. degree in Electrical Engineering at Binghamton University. His areas of research include steganography, steganalysis, digital image forensics, and machine learning.

Jessica Fridrich is Distinguished Professor of Electrical and Computer Engineering at Binghamton University. She received her PhD in Systems Science from Binghamton University in 1995 and MS in Applied Mathematics from Czech Technical University in Prague in 1987. Her main interests are in steganography, steganalysis, and digital image forensics. Since 1995, she has received 20 research grants totaling over $11 mil that lead to more than 180 papers and 7 US patents.