Deep Residual Network for Steganalysis of Digital Images

Mehdi Boroumand, *Student Member, IEEE*, Mo Chen, *Member, IEEE*, and Jessica Fridrich, *Fellow, IEEE*

***Abstract*—Steganography detectors built as deep convolutional neural networks have firmly established themselves as superior to the previous detection paradigm – classifiers based on rich media models. Existing network architectures, however, still contain elements designed by hand, such as fixed or constrained convolutional kernels, heuristic initialization of kernels, the thresholded linear unit that mimics truncation in rich models, quantization of feature maps, and awareness of JPEG phase. In this paper, we describe a deep residual architecture designed to minimize the use of heuristics and externally enforced elements that is universal in the sense that it provides state-of-the-art detection accuracy for both spatial-domain and JPEG steganography. The key part of the proposed architecture is a significantly expanded front part of the detector that "computes noise residuals" in which pooling has been disabled to prevent suppression of the stego signal. Extensive experiments show the superior performance of this network with a significant improvement especially in the JPEG domain. Further performance boost is observed by supplying the selection channel as a second channel.**

***Index Terms*—Steganography, steganalysis, convolutional neural network, deep residual network, selection channel, SRNet**

I. Introduction

Steganography in its modern form is a private, covert communication method in which the sender hides the message inside an innocuous looking cover object using an algorithm driven by a secret shared with the recipient. The communication channel is observed by an adversary or warden who tries to establish whether the communicating parties use steganography. The most popular source of cover objects are digital multimedia files and images in particular. As of 2017, 46% of all steganographic tools available on the Internet can hide messages in digital images stored in raster formats, such as BMP, PNG, and TIFF, and the lossy JPEG format.[1]

With the exception of steganographic schemes based on Least Significant Bit (LSB) replacement [14], [56], [57] and steganography in singular cover sources that permit powerful compatibility attacks [6], [22], [37], [42], the most accurate detectors have been built using the tools of machine learning. This trend has been started by Memon et al. [1], [2] and Farid [18] in early 2000's and was later greatly improved by representing images with higher-order statistics of noise residuals or DCT coefficients [43], [49], [68]. It culminated in what is recognized today as steganalysis with rich models [9], [15], [17], [19], [30], [38], [50], [53] and scalable machine learning [13], [40], [44].

Recently, deep learning [23] has been proposed for steganalysis in an attempt to improve detection accuracy by jointly optimizing the image representation (features) as well as the classifier. Beginning with detectors that used stacked auto-encoders [52], in an early influential work by Qian et al. [45] the authors described a neural network steganalyzer with a Gaussian activation function equipped with a fixed preprocessing high-pass KV filter (Eq. (9) in [39]) whose role was to suppress the image content and thus improve the signal-to-noise ratio between the stego signal and the host image. The authors observed that without the fixed high-pass filter their network did not converge. The XuNet proposed in [61], [62] was the first architecture with a competitive performance. It employed the absolute value layer and TanH activation [23, Ch. 6.3.2, p. 189] in the front part of the network, batch normalization [33], and 1×1 convolutions to compactify the feature maps. It, too, contained a fixed high-pass filter as part of image preprocessing during training and testing. The next advancement, the YeNet [65], can be considered as a breakthrough result as the proposed detector significantly improved upon established steganalysis detectors in the spatial domain. YeNet contained several novel design elements: a new activation function called the Thresholded Linear Unit (TLU), thirty 5×5 kernels in the first layer initialized with SRM (Spatial Rich Model [19]) filters, and an effective way to incorporate the selection channel into the network based on [15], [16]. The work also pointed out the importance of using larger training datasets for deeper networks and the merit of alternative adaptive optimizers, in particular the AdaDelta gradient descend

The work on this paper was partially supported by NSF grant No. 1561446, the Air Force Office of Scientific Research under the research grant FA9950-12-1-0124, and DARPA under agreement number FA8750-16-2-0173. 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.

The authors are with the Department of Electrical and Computer Engineering, Binghamton University, NY, 13902, USA. Email: {mboroum1,fridrich}@binghamton.edu, mochen8@gmail.com

The authors would like to thank Prof. Jiangqun Ni for sharing code and for discussions. Special thanks go to Clement Fuji-Tsang for valuable insight and guidance during his stay at Binghamton University.

[1] N. Johnson, personal communication, 2017.

variant [66]. A deep residual network for steganalysis has recently been proposed in [58]. This work is, unfortunately, faulty, because the detector was essentially trained to recognize, when presented with batches of unlabeled cover-stego pairs, which one of them is cover and which is stego, which is a significantly easier task, unrealistic in any practical application. The authors mentioned on their web site that they were working on a modified architecture.[2]

Detectors constructed using deep learning have also advanced the state of the art in the JPEG domain [10], [60], [64], [67]. Chen et al. [10] modified the XuNet for steganalysis of JPEG images by splitting the feature maps into 64 parallel channels to make the architecture aware of JPEG phase – the underlying grid of 8×8 pixels. The design mimicked the construction of the so-called JPEG-phase-aware noise residuals discovered by Holub et al. [29], [30] and later improved by using Gabor filters for noise residual extraction [50], [59] and making them aware of the selection channel [15]. A 20-layer deep network with shortcut connections [26], [27] for steganalysis of J-UNIWARD [31] has been proposed by Xu [60]. This architecture, too, relied on fixed preprocessing DCT kernels in the first convolutional layer and thresholding its feature maps.

When designing the architecture proposed in this paper, our goal was a clean end-to-end design that could be used for a wider range of applications and work well for steganalysis in both spatial and JPEG domains. We let ourselves be guided by the latest advancements in deep learning and rather general principles and insights to minimize the use of externally enforced constraints or heuristics. Fixed or constrained preprocessing kernels or kernels initialized to SRM filters or DCT bases can in fact be detrimental for the overall network performance depending on the characteristics of the stego signal. High-pass filters, such as the popular KV filter, suppress a major portion of the stego signal introduced by JPEG steganography because the embedding modifications are applied to quantized DCT coefficients. This has already been observed and analyzed in [10] where the authors introduced additional fixed filters into the first convolutional layer to improve detection of JPEG steganography. Ideally, however, the best filters should also be learned rather than enforced as it is unlikely that hand-designed filters or non-random kernel initializations will be optimal for the chosen architecture.

The overall design consists of four different types of layers, two of which involve the so-called residual shortcuts that have been shown in the literature [26], [27] to improve convergence and help learn the parameters in upper layers of deep networks, which are typically the hardest to learn. Functionally, the network consists of three serially connected segments – the front segment whose role is to learn effective "noise residuals," the middle segment that compactifies the feature maps, and the last segment is a simple linear classifier. The front segment consists of seven layers in which pooling [23, Ch. 9.3, pp. 330-334] has been disabled to prevent suppression of the stego signal due to averaging neighboring samples in feature maps during average pooling.

We would like to emphasize that, in its original form, we do not supply the network with the knowledge of the selection channel as we firmly believe that, for the best results, the network should become aware of the selection channel via end-to-end training. Having said this, we acknowledge that introducing the selection channel via a parallel branch in the first layer did improve the performance, which indicates a space for future improvement in the quest for a completely data-driven steganography detector.

At this point, the authors would like to point out a terminology clash between steganalysis and deep learning as the term "residual" has been firmly established in both fields but is used for two completely different entities. To prevent potential confusion, the phrase "noise residual" will be strictly used for a pixel prediction error in steganalysis while "residual layer / module / connection" will always relate to the popular residual network architecture in deep learning [26], [27].

Section II contains the description of the proposed network architecture and a discussion of our design choices. The training, which is unified in both spatial and JPEG domain, is detailed in Section III, where we also describe the setup of all our experiments, the performance evaluation metric as well as the list of prior art with which the proposed detector is compared. The results of experiments in spatial and JPEG domain appear in Section IV. The performance is evaluated in terms of the minimal detection error under equal priors. We also report the detection performance on selected cases using the receiver operating characteristic curves with the false-alarm rate for true positive rates of 0.5 and 0.3. In Section V, we show that further boost of detection accuracy can be achieved in both domains by introducing the selection channel into the network. The paper is closed in Section VI with a discussion of potential further improvements and our anticipated future effort.

II. SRNet for image steganalysis

The proposed network architecture is called SRNet – Steganalysis Residual Network. The word "residual" refers to both the central term used in steganalysis and residual layers with shortcut connections from deep learning [26]. The shortcut connections help propagate gradients to upper layers, which are the hardest to train because of the vanishing gradient phenomenon [21] that often negatively affects the convergence and performance of deep architectures [26], [27]. They also encourage feature reuse in the training process. We first describe the architecture of SRNet and then explain and justify each component separately, motivating thus the design.

A. Architecture

Although it is not generally possible to claim that a certain part of a network detector executes a specific task, we

[2]https://github.com/Steganalysis-CNN/residual-steganalysis

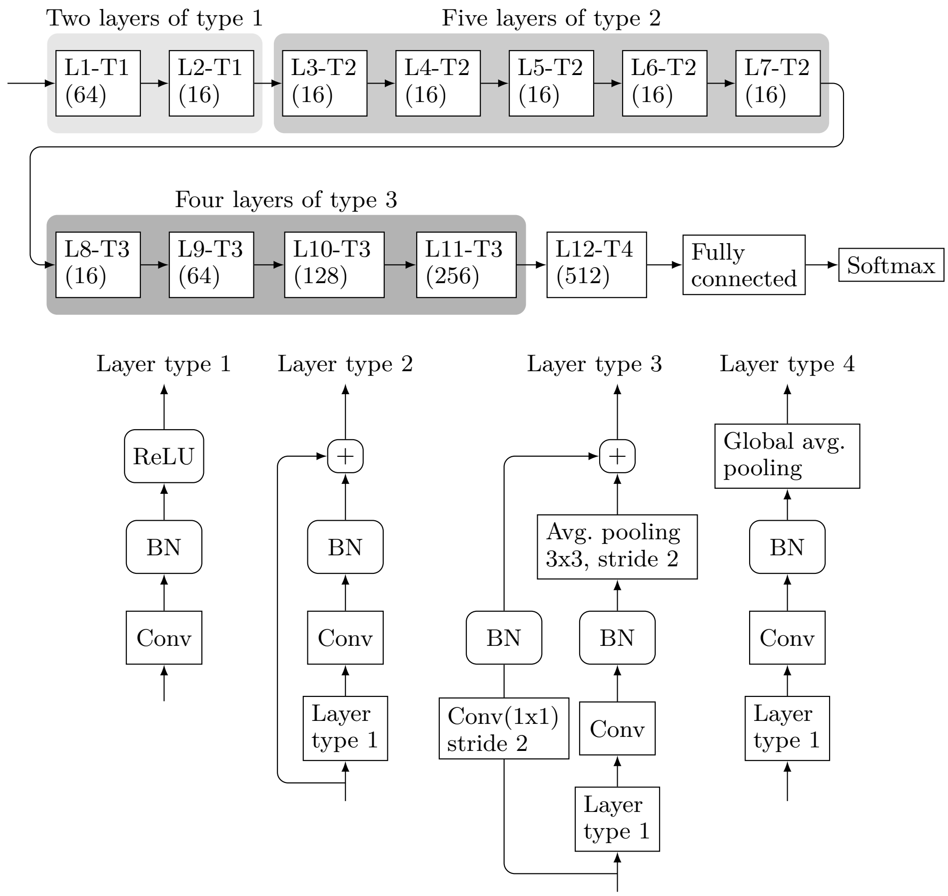


Figure 1. Architecture of the proposed SRNet for steganalysis. The first two shaded boxes correspond to the segment extracting noise residuals, the dark shaded segment and Layer 12 compactify the feature maps, while the last fully connected layer is a linear classifier. The number in the brackets is the number of 3×3 kernels in convolutional layers in each layer. BN stands for batch normalization.

found it useful to view the proposed detector schematically depicted in Figure 1 as a concatenation of three segments: the front segment responsible for extracting the noise residuals, outlined in the figure by the first two shaded segments (Layers 1–7), the middle segment whose goal is to reduce the dimensionality of the feature maps, the third shaded segment and Layer 12, and the last segment, which is a standard fully connected layer followed by a softmax node [23], the linear classifier.

The input is assumed to be a grayscale 256×256 image.[3] All convolutional layers employ 3×3 kernels and all non-linear activation functions are ReLU. Note that Layers 1–7 use unpooled feature maps on their input. Pooling in the form of 3×3 averaging with stride 2 is applied on the output of Layers 8–11. In Layer 12, 512 feature maps of dimension 16×16 are reduced to a 512-dimensional feature vector by computing statistical moments (averages) of each 16×16 feature map. This 512-dimensional output enters the classifier part of the network. The first two layers do not contain any residual shortcuts or pooling. Layers 3–7 have residual shortcuts and no pooling. Layers 8–11 contain both pooling and residual shortcuts.

[3]Reference [20] explains how to steganalyze images of arbitrary size with network detectors.

SRNet contains two types of layers with shortcuts because unpooled layers (Type 2) require different shortcut connections than pooled layers (Type 3). The first two layers of Type 1 with 3×3 filters worked better for us than one layer with 5×5 filters. Their purpose is to begin with a larger number of kernels (64) and then decrease the number of feature maps to 16 before the unpooled layers to save on memory. The Type 4 layer is different from the last layer of Type 3 because of the global pooling applied before the fully connected classifier part.

B. *Motivating the architecture*

The key part of the SRNet is the noise residual extraction segment consisting of the first seven layers. Because average pooling is a low-pass filter, it reinforces content and suppresses noise-like stego signals by averaging adjacent embedding changes. While this is desirable in typical computer vision applications for classifying content, it is detrimental for steganalysis where the signal of interest is the stego noise while the "noise" is the image content. Guided by this insight, SRNet does not use pooling until Layer 8 to avoid decreasing the energy of the stego signal and allow it to optimize the noise residual extraction process for various types of selection channels and steganographic embedding changes.

All filters in SRNet are randomly initialized and learned via an end-to-end training process. This allows the network to adapt to a greater variety of stego signals because the polarity of and dependencies among embedding changes vary significantly across different steganographic methods and especially domains. Embedding modifications introduced by spatial-domain embedding methods that minimize an additive distortion, such as WOW [28], HILL [41], S-UNIWARD [31], and MiPOD [47] are largely uncorrelated, while changes to quantized DCT coefficients in JPEG image steganography lead to a stego signal with significant energy in low and medium spatial frequencies.

The proposed architecture was formed based on results of many experiments in which we tested different allocations of resources to the three above mentioned segments so that the network can be trained with a reasonable minibatch size on a single GPU with 12 GB of memory, examples of which are the popular Titan X and Xp, Tesla K40 and K80, and GTX 1080 Ti (11 GB). Most of the exploration focused on determining the number of layers in each segment, the number of filters in each layer, and the optimizer.

The remainder of this section is divided into subsections, each devoted to a specific design element of SRNet. The experimental results quoted here were all obtained with the setup explained in Section III on the standardized dataset BOSSbase + BOWS2 (Section III-A) with the detector accuracy reported using the minimal total detection error P_{E} under equal priors based on the training, validation, and testing (Section III-D).

1) Activations: Besides the ReLU, we have also experimented with TanH activation, the leaky ReLU, ELU [11], and SELU [36], but they did not bring any performance gain. To avoid additional complexity and guided by simplicity, we selected ReLU for all activation functions in our network.

Note that layers of Type 2 and 3 do not use ReLU after the shortcut connections. While the original residual networks [26], [27] do include ReLU after the addition of the shortcut connections, with these activations removed, we observed a small gain of up to 1% in detection accuracy.

2) Residual shortcuts: To assess the importance of shortcut connections in SRNet, we removed them from layers of Type 2 and 3 and observed the change in detection accuracy. For example, for HILL at 0.1 and 0.2 bpp the loss of classification accuracy was about 0.5% and for J-UNIWARD at 0.4 bpnzac, quality factor 95, the loss was 1.5%. While the performance in these cases was still competitive, the loss of detection power increased with decreased class separability, e.g., for small payloads and larger JPEG quality.

3) Dense connections and inception: Dense connections in deep learning were introduced with a similar goal as residual layers – to help with gradient propagation and convergence, feature reuse, and to reduce the number of parameters to learn [32]. We investigated the effect of dense connections introduced in the second segment of the SRNet – unpooled Layers 3–7. On experiments with the embedding algorithms HILL and S-UNIWARD at 0.4 bpp, the SRNet with dense connections did not provide statistically significant better results as SRNet with residual connections (the statistical significance was assessed based on the statistical spread of detection accuracy w.r.t. the snapshot selected for the final detector). Dense connections, however, may have more impact on deeper architectures than the SRNet.

The main idea behind “inception” is that each layer concatenates the outputs of filters of different sizes, which is reminiscent of fusing multiple-resolution representations in image processing [51]. Type 3 layers in SRNet (see Figure 1) sum the outputs of what is an effective 5×5 filter in the main branch (in terms of the receptive field) and a 1×1 filter (the shortcut branch). We added an additional branch to this layer type with 3×3 filters followed by batch normalization. This required other changes in the architecture to fit the modified SRNet in GPU memory – we decreased the number of feature maps in Type 3 layers to one half. SRNet modified in this manner gave a slightly worse (0.5–1%) detection accuracy on both HILL and S-UNIWARD tested at 0.4 bpp. Due to limited GPU memory, a proper study of inception modules within the SRNet would require a comprehensive study that is beyond the scope of this paper.

4) Unpooled layers: We now comment on the number of unpooled layers and their effect on detection. Decreasing their number from seven to six or five while keeping the rest of the architecture unchanged lead to a small and gradual loss of accuracy. For example, for J-UNIWARD at 0.4 bpnzac (bits per non-zero AC DCT coefficient) and JPEG quality 75, the detection error P_{E} increased from 0.0670 to 0.0701 and 0.0748 when the number of unpooled layers was changed from 7 to 5 and 4, respectively. This loss increases with decreasing payload. Also, we observed that this loss is typically smaller in the spatial domain and larger in the JPEG domain. Across the tested algorithms in both domains, the detection accuracy tends to level out at 5–6 unpooled layers. We opted for seven in our proposed design to avoid potential loss of detection for more diverse cover and stego sources.

To assess the significance of disabling pooling in Layers 1–7, we carried out additional experiments in which pooling has been progressively enabled in Layers 7, 6, 5, and 4. Note that enabling pooling in more than four layers would require removing layers from group 3 because the size of the feature maps before the output layer decreases from 16×16 to 8×8, and eventually 1×1 when pooling is enabled in four layers.

The experiments were executed for HILL at 0.4 bpp and J-UNIWARD at 0.4 bpnzac to cover both embedding domains. With enabling average pooling in Layers 7–4, starting with Layer 7, the detection error for HILL rapidly increased from 0.1414 (with the original SRNet) to 0.1528, 0.1823, 0.2202, and 0.3697. For J-UNIWARD, the detection error grew from 0.0670 to 0.0755, 0.0886, 0.1263, and 0.1710.

5) Number of filters: The effect of the number of filters in the first layer has a larger impact in the JPEG domain than in the spatial domain. While the detection error, P_{E}, for HILL at 0.4 bpp increased negligibly when using only 32 and 16 filters instead of 64 in the first layer (0.1414, 0.1432, and 0.1438 for 64, 32, and 16 filters), for J-UNIWARD at 0.1 bpnzac at JPEG quality 75, decreasing the number of filters from 64 to 32 lead to an increase of P_{E} of about 1%. Increasing the number of filters beyond 64 did not seem to lead to any improvement in detection.

6) Optimizer: Finally, we experimented with several optimizers, including the AdaDelta [66], Adam [35], Adamax [35], and a simple stochastic gradient descend [23, Ch. 8.3.1, pp. 286-288]. In the end, we settled on Adamax since it provided the most reliable and fastest convergence.

III. Setup of experiments

This section describes the common core of all experiments that appear in Section IV and V, including the datasets and SRNet training, the list of prior art to which SRNet is to be compared, and the evaluation metric.

A. Datasets

SRNet was primarily evaluated and contrasted with prior art on the union of BOSSbase 1.01 [3] and BOWS2 [4], each containing 10,000 grayscale images resized from their original size 512×512 to 256×256 using `imresize` with default setting in Matlab. For JPEG experiments, this source was additionally compressed with quality factors 75 and 95.

Randomly chosen 4,000 images from BOSSbase and the entire BOWS2 dataset were used for training with 1,000 BOSSbase images set aside for validation. The remaining 5,000 BOSSbase images were used for testing. This setup permitted a direct comparison with the current state-of-the-art spatial-domain detector, the YeNet [65]. In summary, $2 \times 14,000$ cover and stego images were used for training, $2 \times 1,000$ for validation, and $2 \times 5,000$ for testing. This applies to both the spatial and JPEG domain and all network detectors. JPEG images were decompressed without rounding to integers.

To test the network on a significantly larger and more realistic dataset, we performed additional experiments on ImageNet, namely its CLS-LOC version [46] containing 1,281,167 JPEG images meant to be used for training sorted into 1,000 categories (the dataset used in [60]). We selected 250 images from each category at random, subjecting each image that was larger than 256×256 pixels and whose JPEG quality was above 75 to the following chain of processing in Matlab: decompression to the spatial domain (`imread`), cropping the upper left tile of size 256×256, conversion to grayscale using `rgb2gray`, and recompression with JPEG quality factor 75. This mimics the preprocessing that was executed in [60], [67]. In particular, the requirement to work only with JPEG images with quality larger than 75 was imposed to avoid working with images exhibiting traces of double compression (lower quality followed by larger quality) as this would introduce peaks and valleys in histograms of quantized DCT coefficients, which could be exploited for targeted attacks. The total size of this dataset was thus 2×250,000 cover-stego images out of which 2×10,000 pairs were selected for validation and 2×40,000 for testing.

B. SRNet training

The SRNet has been trained in both domains with the same hyperparameters and in the same fashion. The stochastic gradient descend optimizer Adamax[4] [35] was used with minibatches of 16 cover-stego pairs. The training database was shuffled after each epoch. Images in each batch were subjected to data augmentation with random mirroring and rotation of images by 90 degrees. The batch normalization parameters were learned via an exponential moving average with decay rate 0.9. The filter weights were initialized with the He initializer[5] and 2×10^{-4} L2 regularization. The filter biases were set to 0.2 and no regularization. For the fully connected classifier layer, we initialized the weights with a zero mean Gaussian with standard deviation 0.01 and no bias.

On our dataset, the training was run for 400k iterations (457 epochs) with an initial learning rate of $r_1 = 0.001$ after which the learning rate was decreased to $r_2 = 0.0001$ for an additional 100k iterations (114 epochs). The snapshot achieving the best validation accuracy in the last 100k iterations was taken as the result of training. This training strategy was applied for all embedding algorithms for payload 0.4 bpp/bpnzac (bits per pixel / bits per non-zero AC DCT coefficient) with the exception of J-UNIWARD at JPEG quality 95 (see the next paragraph). The detectors for all remaining payloads were built via curriculum training [5] with 50–100k iterations (57–114 epochs) with learning rate r_1 and an additional 50k iterations (57 epochs) with the smaller learning rate r_2. Again, the best validation snapshot in the last 50k iterations was taken as the detector. While this was applied in both spatial and JPEG domain, we observed that in the spatial domain the same results could be obtained by curriculum training only with the smaller learning rate.

For J-UNIWARD and JPEG quality factor 95 at 0.4 bpnzac, we experienced convergence problems when training from a randomly initialized network. This was resolved by seeding the network with the detector trained for J-UNIWARD for quality factor 75 at 0.4 bpnzac, after which we trained for 400k iterations with learning rate r_1 followed by 100k iterations with r_2.

We tested two types of curriculum training – by seeding with the network trained for payload 0.4 bpp/bpnzac and by training in a progressive manner that is perhaps best described symbolically as 0.1←0.2←0.3←0.4→0.5. In other words, first the detectors for payload 0.3 and 0.5 were trained by seeding with the network trained for 0.4.

[4]Code available from https://github.com/openai/iaf/blob/master/tf_utils/adamax.py

[5]https://arxiv.org/pdf/1502.01852v1.pdf

Then, the detector for payload 0.2 was seeded with the network trained for 0.3, etc. While both types of curriculum training gave similar results in the spatial domain, the progressive training gave slightly better results in the JPEG domain.

C. Tested prior art

For comparison with the current state of the art on the union of BOSSbase and BOWS2, in the spatial domain SRNet was compared with YeNet [65] and on JPEG algorithms with the PNet / VNet [10] and the network recently proposed by Xu [60], which we call in this paper J-XuNet to distinguish it from the network introduced in [62]. We note that when we attempted to train the YeNet on decompressed JPEGs with quality factor 75 embedded with J-UNIWARD at 0.4 bpnzac the network did not appear to converge.

To show the gain in detection accuracy w.r.t. the old detection paradigm based on the ensemble classifier and rich models, we steganalyzed all spatial-domain embedding algorithms with the maxSRMd2 [17] features non-linearly normalized using random conditioning (RC) [8]. JPEG steganography was steganalyzed with the Selection-Channel-Aware Gabor Filter Residuals [15] (SCA-GFR). The SCA-GFR features were not normalized or transformed [7], [8] because this type of features does not benefit from such preprocessing.

All prior art CNN detectors were trained as described in the corresponding papers. We observed that for the J-XuNet on 256×256 images, it was beneficial to decrease the learning rate by 10% every 8 epochs instead of 16 to avoid a loss of performance for small payloads. For J-UNIWARD quality factor 95, we had to train the J-XuNet for payloads 0.1 and 0.2 bpnzac via curriculum training from 0.3 bpnzac.

Due to the size of ImageNet, we limited our experiments on this dataset to J-UNIWARD at quality factor 75 and only compared to J-XuNet and the recently proposed hybrid deep network incorporating J-XuNet as a “subnet” as described in Sec. III.E of [67] (Fig. 13a), which we abbreviate in this paper as H-Net.

All detectors were trained on exactly the same data sets as the SRNet, implemented in TensorFlow, and run on a single GPU. It takes approximately two and half days to train the SRNet on a Titan Xp GPU. Note that we did not form ensembles of CNN detectors in this paper. Quite likely, further small improvement in detection accuracy could be obtained across all investigated network detectors by forming an ensemble either over different snapshots obtained from a single training or over independently trained networks.

D. Evaluation metric

The detection performance was measured with the total classification error probability on the testing set under equal priors $P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2}(P_{\mathrm{FA}} + P_{\mathrm{MD}})$, where P_{FA} and P_{MD} are the false-alarm and missed-detection probabilities. For selected cases, we show the ROC curves and an alternative measure of performance, the false-alarm rates for stego-image detection probability $P_{\mathrm{D}} = 1 - P_{\mathrm{MD}} = 0.5$ and 0.3.

Table I

Detection error P_{E} for maxSRMd2 with random conditioning and ensemble, SRNet, and selection-channel-aware YeNet for five payloads in bpp and three spatial domain embedding algorithms.

	Detector	0.1	0.2	0.3	0.4	0.5
S-UNI	maxSRM+RC	.3817	.2904	.2223	.1783	.1429
	SCA-YeNet	.3220	.2224	.1502	.1281	.1000
	SRNet	**.3104**	**.2090**	**.1432**	**.1023**	**.0705**
HILL	maxSRM+RC	.3768	.3168	.2707	.2338	.1855
	SCA-YeNet	.3380	.2538	.1949	.1708	.1305
	SRNet	**.3134**	**.2353**	**.1830**	**.1414**	**.1151**
WOW	maxSRM+RC	.2998	.2144	.1684	.1350	.1122
	SCA-YeNet	**.2442**	.1691	.1229	.0959	.0906
	SRNet	.2587	**.1676**	**.1197**	**.0893**	**.0672**

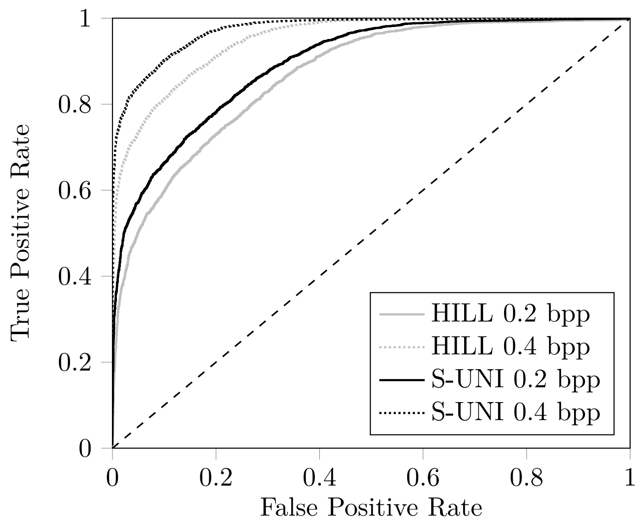
The results reported in the next section are for one random 50/50 split of BOSSbase because it would not be computationally feasible to train all networks on multiple different splits to obtain a more statistically robust result. To assess the statistical spread across different BOSSbase splits and thus interpret the statistical significance of the improvement of SRNet w.r.t. the state of the art, we trained the SRNet on five different 50/50 BOSSbase splits (BOWS2 was always a part of the training set) for HILL at 0.3 bpp and J-UNIWARD at 0.4 bpnzac and JPEG quality 75. The standard deviation of P_{E} across these five splits was 0.0035 and 0.0016, respectively. The statistical spread appears coincidentally comparable to what has typically been reported for detectors implemented with rich models and the ensemble classifier (see, e.g., [15], [17]).

IV. Experiments

This section contains the results of all experiments and their interpretation divided into two subsections based on the type of the embedding domain.

A. Spatial domain

For spatial domain steganalysis, we report the results for five payloads: 0.1–0.5 bpp (bits per pixel) for WOW [28], HILL [41], and S-UNIWARD [31]. The detection error P_{E} is shown in Table I. Depending on the algorithm and payload SRNet improves upon SCA-YeNet by up to 3% in P_{E}. The biggest improvement is typically observed for larger payloads. The only exception is for WOW for the smallest tested payload 0.1 bpp when SRNet performs by 1.5% worse than the SCA-YeNet. This loss of performance is due to the fact that SRNet does not make explicit use of the selection channel while YeNet benefits quite significantly by employing the selection channel for WOW (c.f. columns 3 and 5 in Table VIII in [65]). In Section V, we show that this loss can be compensated by introducing the selection channel to SRNet in a similar manner as in YeNet. Finally, both network detectors clearly outperform the old steganalysis paradigm.



		SRNet		maxSRM	
Embedding	bpp	$P_{\mathrm{FA}}(0.5)$	$P_{\mathrm{FA}}(0.3)$	$P_{\mathrm{FA}}(0.5)$	$P_{\mathrm{FA}}(0.3)$
S-UNI	0.2	.0222	.0032	.1244	.0336
	0.4	.0006	.0002	.0200	.0034
HILL	0.2	.0488	.0100	.1608	.0436
	0.4	.0042	.0010	.0482	.0118

Figure 2. ROC curves of SRNet for S-UNIWARD and HILL at 0.2 and 0.4 bpp together with two detection performance measures: P_{FA} for $P_{\mathrm{D}} = 0.5$ and 0.3 also computed for the low-complexity linear classifier with the maxSRMd2 feature set transformed using random conditioning.

ROC curves for rich-model based detectors are well known to be mean-shifted Gauss-Gauss (see, e.g., [12]) and as such do not perform well for low false alarms. In contrast, the detection statistic outputted by network detectors exhibits non-Gaussian characteristics and, as we found out, achieves *significantly* better performance for low rates of false alarm, a goal identified as one of the most relevant problems for practitioners in [34]. Figure 2 shows four ROC curves of SRNet for S-UNIWARD and HILL for two payloads and the false alarm rates P_{FA} for two test powers: $P_{\mathrm{D}} \in \{0.3, 0.5\}$. For the larger payload 0.4 bpp, $P_{\mathrm{D}} = 0.5$ can be achieved with $P_{\mathrm{FA}} = 6 \times 10^{-4}$ for S-UNIWARD and 4.2×10^{-3} for HILL. In contrast, the low-complexity linear classifier [12] with maxSRMd2 features [17] (the last two columns in the table underneath Figure 2) exhibits 4–30 times larger (!) false alarms for the two test powers.[6]

B. Transfer learning

To assess the ability of the SRNet to detect mismatched stego sources, which is a situation likely to be encountered in practice, we include the result of an investigation in which the SRNet was trained on one embedding algorithm and tested on a different one at the same payload. Table II shows that the SRNet trained on the least detectable algorithm (MiPOD) transfers the best while, when trained on the most detectable algorithm (WOW), it transfers the least. This is consistent with the results reported in [10] for the JPEG-phase-aware network in JPEG domain.

Table II
DETECTION ERROR P_{E} FOR SRNET TRAINED ON ONE ALGORITHM AND TESTED ON OTHER ALGORITHMS. PAYLOAD FIXED AT 0.4 BPP.

TRN\TST on	WOW	HILL	S-UNI	MiPOD
WOW	.0893	.3228	.1552	.2879
HILL	.1742	.1414	.2742	.2180
S-UNI	.1102	.2483	.1023	.2116
MiPOD	.1476	.1888	.1596	.1497

C. JPEG domain

For the JPEG domain, J-UNIWARD [31] and UED-JC [25] for payloads 0.1–0.5 bpnzac (bits per non-zero AC DCT coefficient) were tested for quality factors 75 and 95. The results of the experiments are shown graphically in Figures 4 and 5 and Table III.

For UED-JC, SRNet detection error is up to 8% lower than J-XuNet and the improvement is up to 17% w.r.t. state of the art for J-UNIWARD for quality factor 95. A very significant improvement of up to 18.5% (!) is observed w.r.t. the old detection paradigm, the SCA-GFR with ensemble classifier. Four ROC curves of the SRNet are shown in Figure 6 for J-UNIWARD, payloads 0.2 and 0.4 bpnzac, and two quality factors. Again, the network detector enjoys a very small false alarm rate for probability of detection 0.5 and 0.3 that is significantly smaller than for the old detection paradigm, the low-complexity linear classifier [13] with the SCA-GFR feature set. Note that for J-UNIWARD at 75 JPEG quality and payload 0.4 bpnzac, detection rate $P_{\mathrm{D}} = 0.3$ was achieved with no false alarms on the 10,000 images from the testing set.

Figure 3 shows an example of the progression of the training and validation error when training SRNet on J-UNIWARD at 75 JPEG quality and payload 0.4 bpnzac. Note the drop in detection error due to decreasing the learning rate at iteration 400k.

1) ImageNet: A subset of the CLS-LOC version of the ImageNet [46] with 250,000 grayscale 256×256 JPEG images was included in our tests to show the performance of SRNet on a more realistic dataset containing images from a large number of different sources including multiple-compressed images. For comparison with prior art, we included J-XuNet and the hybrid network with the "J-XuNet model" by Zeng et al. [67] (Fig. 13a) that has been published during the writing of this paper.

The detection error P_{E} as a function of embedded payload size for J-UNIWARD at JPEG quality 75 is shown in Figure 7. The gain of SRNet w.r.t. both J-XuNet and H-Net is between 5–7%. Also, in contrast to the claims made in [67], H-Net with J-XuNet as the subnet provides approximately the same performance as J-XuNet itself.

[6]The low-complexity linear classifier was used instead of the ensemble to be able to obtain the performance measures reported in Figure 2.

Table III
Detection error P_E for SRNet and prior art for five payloads in bpnzac for J-UNIWARD and UED-JC for quality factors 75 (left) and 95 (right).

Embedding Method	Detector	QF 75 0.1	0.2	0.3	0.4	0.5	QF 95 0.1	0.2	0.3	0.4	0.5
J-UNIWARD	SCA-GFR	.4197	.3257	.2400	.1728	.1190	.4798	.4430	.3951	.3448	.2916
	VNet	.4029	.2928	.1938	.1258	.0815	.4756	.4373	.3898	.3304	.2668
	PNet	.3917	.2904	.1966	.1283	.0799	.4741	.4253	.3751	.3182	.2435
	J-XuNet	.4310	.2849	.1895	.1207	.0776	.4812	.4512	.4146	.3232	.2243
	SRNet	**.3201**	**.1889**	**.1153**	**.0670**	**.0385**	**.4277**	**.3440**	**.2516**	**.1762**	**.1148**
UED-JC	SCA-GFR	.3176	.2154	.1381	.0871	.0579	.4376	.3700	.2995	.2390	.1859
	VNet	.2565	.1352	.0770	.0433	.0223	.4131	.3316	.2498	.1867	.1254
	PNet	.2470	.1290	.0740	.0420	.0218	.3957	.3095	.2245	.1617	.1015
	J-XuNet	.2144	.0972	.0508	.0287	.0163	.3848	.2979	.1991	.1292	.0883
	SRNet	**.1311**	**.0568**	**.0285**	**.0188**	**.0093**	**.3044**	**.2028**	**.1261**	**.0877**	**.0500**

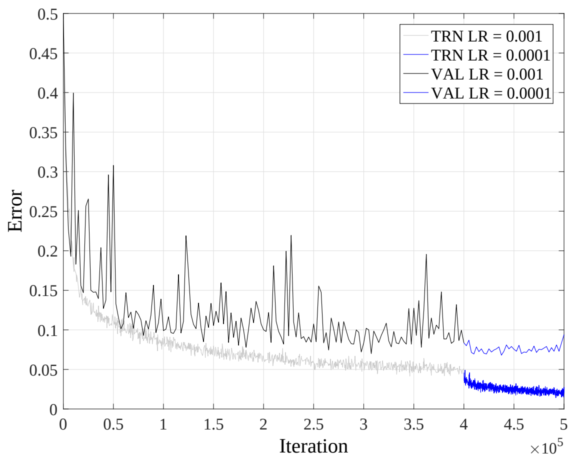


Figure 3. Training and validation error P_E for J-UNIWARD QF 75 at 0.4 bpnzac.

This was also observed on our other dataset, BOSSbase + BOWS2, but is not shown in this paper.

It is also interesting to contrast the detection errors on ImageNet with those on the more "sand boxed" environment – the union of BOSSbase and BOWS2. Because of the far greater diversity of ImageNet, the detection error on this source is larger by 6.6–9% compared to the more homogeneous image source.

V. SRNet with selection channel

When detecting a known content-adaptive steganographic algorithm, the Warden may use the fact that the embedding change probabilities (the so-called selection channel) with which the pixels or DCT coefficients were changed in a stego image are known [15]–[17], [53], [54]. Even though the selection channel computed from the stego image will inevitably be different than the selection channel used for embedding by the sender computed from the cover image, these differences are typically fairly small because the selection channel (the embedding costs) are typically insensitive to embedding changes [48]. Furthermore, it has been shown [48] that, at least for classifiers trained with rich media models, it is still beneficial to use an imprecise selection channel (e. g., because the payload size is not known) than not use it at all.

While the incorporation of the selection channel helps detection, it has always been achieved in some heuristic manner. In the so-called t-SRM proposed by Tang et al. [53], the four-dimensional SRM co-occurrence matrices of noise residuals were computed from a subset corresponding to pixels with the largest embedding change probabilities. In maxSRM [17] (and in [54]), the co-occurrences contain the accumulated maximum change rate (the sum of change rates) over all adjacent four-tuples of noise residuals. This idea was further refined in [15], [16] where the authors showed that further improvement can be obtained by replacing the change rate as the quantity being accumulated with an upper bound on the L_1 noise residual distortion due to embedding.

With the introduction of deep learning to steganalysis, researchers investigated various ways how to inform the neural network about the embedding change probabilities [63], [65]. While SRNet was intentionally designed to be free of such heuristic elements to allow a clean end-to-end training and while we believe that a sufficiently complex and suitably designed and trained architecture will not need an external insertion of the selection channel, the SRNet may still benefit from being informed about the probabilistic impact of embedding. Indeed, the experiments in Section IV-A indicate a small loss of detection accuracy w.r.t. SCA-YeNet for small payloads for WOW.

The selection channel has been incorporated in SRNet in the same fashion as in YeNet [65], which was inspired by [16]. We first describe the modification of the architecture for the spatial domain. Given the l-th, $l = 1, \ldots, 64$, convolution kernel $\mathbf{W}^{(l)} \in \mathbb{R}^{3\times 3}$ from the first layer, the convolution $\mathbf{W}^{(l)} \star \mathbf{x}$ is a form of noise residual. The impact of embedding on this noise residual can be quantified by evaluating an upper bound on the L_1 distortion, which, for steganography that modifies cover pixels independently by ± 1, can be computed as $|\mathbf{W}^{(l)}| \star \boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is the matrix of change rates, the selection channel.[7] This bound is added to the feature maps outputted by the first layer to *reinforce*

[7] β_{ij} is the probability of modifying cover element x_{ij}. Thus, for embedding schemes that modify cover values by 1 or -1, β_{ij} is the sum of the two probabilities of changing by 1 and -1.

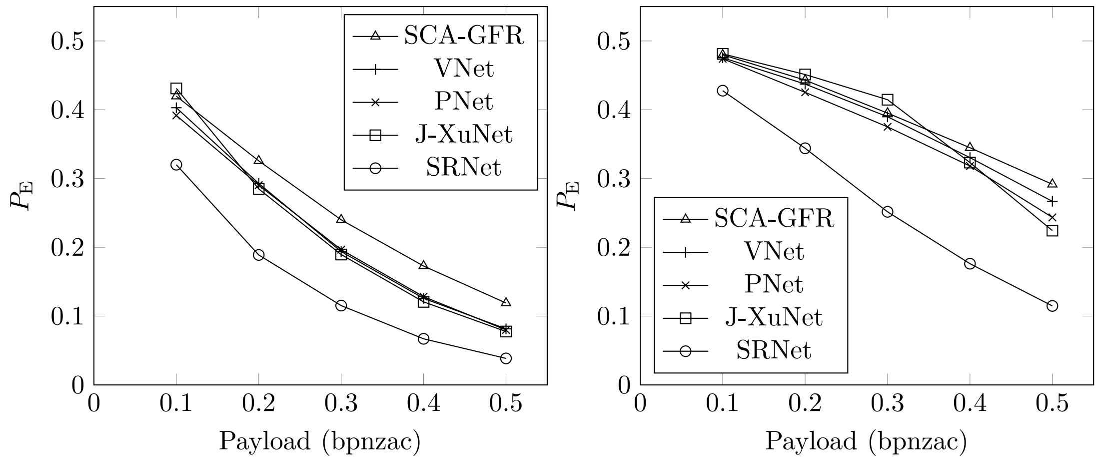


Figure 4. Detection error P_{E} for VNet, PNet, J-XuNet, and SRNet for J-UNIWARD QF 75 (left) and 95 (right).

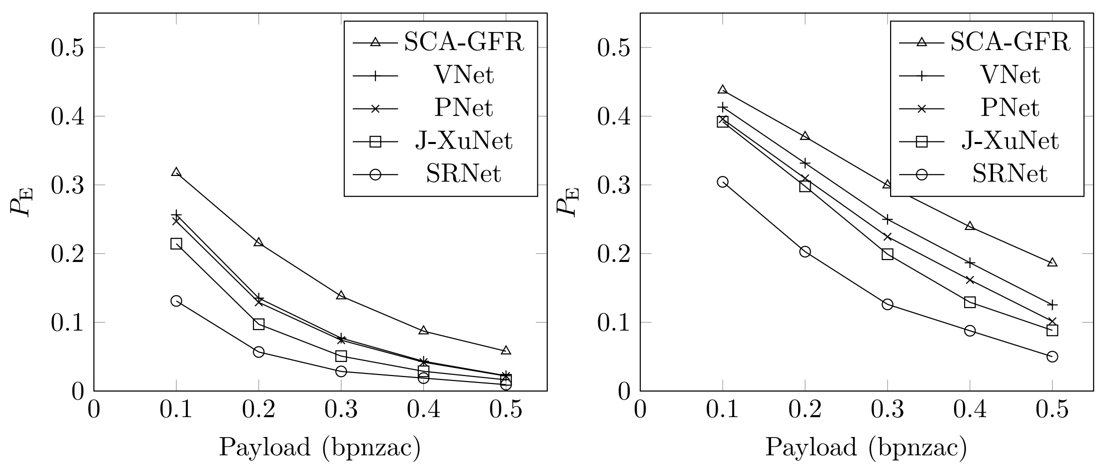


Figure 5. Detection error P_{E} for VNet, PNet, J-XuNet, and SRNet for UED-JC QF 75 (left) and 95 (right).

the output of neurons that are most affected by embedding. The rest of the SCA-SRNet architecture is identical to SRNet (see Figure 1) with the first layer shown in Figure 8. Note that the batch normalization was removed from the first layer to make sure both signals that are added are of similar scale. The kernels applied to the image in the first layer and those applied to the change rates are forced to be the same, e.g., the absolute values of the kernels are merely copied from the main network.

Formally, for the spatial domain, with the $M \times N$ matrices of pixel values $\mathbf{x} = (x_{ij})$ and embedding change probabilities $\boldsymbol{\beta} = (\beta_{ij})$, the lth feature map, $l = 1, \ldots, 64$, that enters the second convolutional layer in the forward pass is

$$\mathrm{ReLU}(\mathbf{W}^{(l)} \star \mathbf{x}) + |\mathbf{W}^{(l)}| \star \boldsymbol{\beta}, \tag{1}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{3\times 3}$ is the lth convolutional kernel from the first layer of SRNet and $'\star'$ denotes the convolution. During learning, the weight vectors in the main branch of the network are copied to the selection-channel branch where the absolute value operation is applied and the network is trained as before.

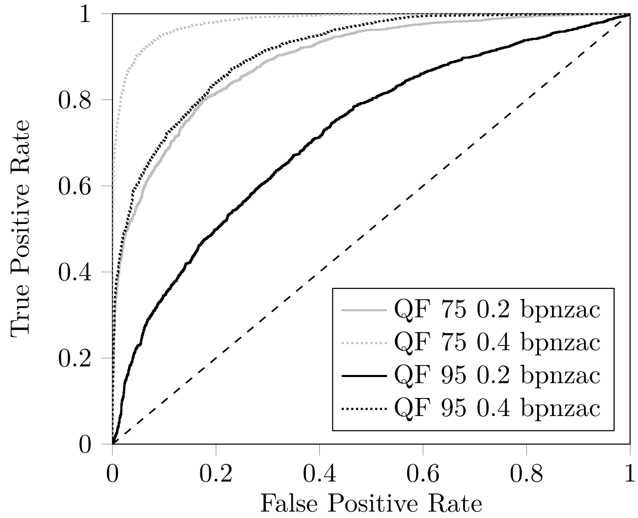
For JPEG domain algorithms, the selection channel is incorporated in a similar fashion. The embedding change probabilities, however, relate to the quantized DCT coefficients rather than pixels. Thus, as the first step, we compute the impact of embedding on pixels as an upper bound t on the L_1 embedding distortion as in Eqs. (18–19) in [15]. This bound in the (a, b)-th JPEG 8×8 block, $0 \le a \le M/8 - 1$, $0 \le a \le N/8 - 1$ is computed as:

$$t_{ij}^{(a,b)} = \sum_{k,l=0}^{7} |f_{ij}^{(k,l)}| q_{kl} \beta_{kl}^{(a,b)}, 0 \le i, j \le 7, \tag{2}$$

where $\beta_{kl}^{(a,b)}$, $0 \le k, l \le 7$, is the change rate corresponding to DCT mode k, l in (a, b)-th DCT 8×8 block, q_{kl} is the JPEG luminance quantization step, and

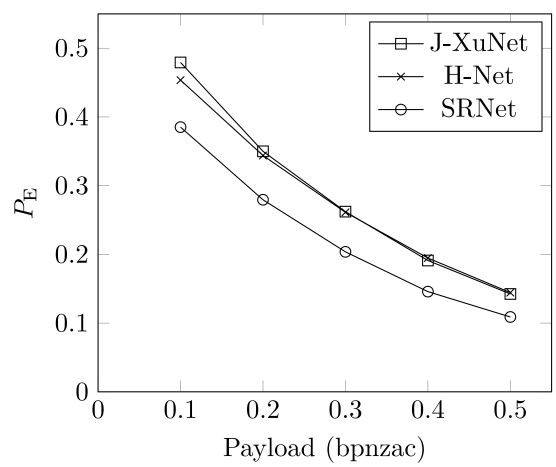
$$f_{ij}^{(k,l)} = \frac{w_k w_l}{4} \cos \frac{\pi k(2i+1)}{16} \cos \frac{\pi l(2j+1)}{16}, \tag{3}$$

$w_0 = 1/\sqrt{2}$ and $w_k = 1$ for $k > 0$, are the coefficients of the DCT. The computation of the matrix $\mathbf{t}$ is a mere *preprocessing* of the change rates and can be done *outside of the network*. The bound $\mathbf{t}$ enters the selection-channel branch in the first layer as shown in Figure 8. Finally, the



		SRNet		SCA-GFR	
QF	bpp	$P_{\mathrm{FA}}(0.5)$	$P_{\mathrm{FA}}(0.3)$	$P_{\mathrm{FA}}(0.5)$	$P_{\mathrm{FA}}(0.3)$
75	0.2	.0296	.0048	.1810	.0726
	0.4	.0008	.0000	.0250	.0062
95	0.2	.2016	.0760	.3954	.2172
	0.4	.0252	.0040	.2058	.0834

Figure 6. ROC curves of SRNet for J-UNIWARD at 0.2 and 0.4 bpp for quality factors 75 and 95 together with two detection performance measures: P_{FA} for $P_{\mathrm{D}} = 0.5$ and 0.3 for the low-complexity linear classifier with the SCA-GFR feature set.



Payload	0.1	0.2	0.3	0.4	0.5
J-XuNet	.4793	.3500	.2622	.1911	.1424
H-Net	.4536	.3439	.2612	.1947	.1444
SRNet	**.3852**	**.2795**	**.2037**	**.1458**	**.1089**

Figure 7. Detection error P_{E} for J-XuNet, H-Net, and SRNet for J-UNIWARD QF 75 on ImageNet.

lth feature map, $l = 1, \ldots, 64$, outputted by the first layer is thus

$$\mathrm{ReLU}(\mathbf{W}^{(l)} \star \mathbf{x}) + \sqrt{|\mathbf{W}^{(l)}| \star \boldsymbol{t}}, \tag{4}$$

where $\mathbf{x}$ is the decompressed JPEG image without rounding to integers. The square root non-linearity is there to

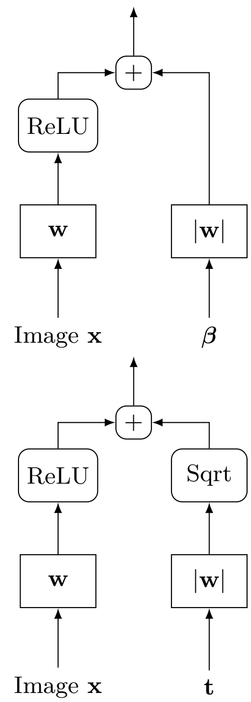


Figure 8. The first layer in SCA-SRNet. The left branch is the main branch while the branch on the right brings in the information about the selection channel. Top: spatial domain, Bottom: JPEG domain.

Table IV
Effect of introducing the selection channel into SRNet (spatial domain).

	Detector	0.1	0.2	0.3	0.4	0.5
S-UNI	SCA-YeNet	.3220	.2224	.1502	.1281	.1000
	SRNet	.3104	.2090	.1432	.1023	.0705
	SCA-SRNet	**.2969**	**.1918**	**.1309**	**.0935**	**.0667**
HILL	SCA-YeNet	.3380	.2538	.1949	.1708	.1305
	SRNet	.3134	.2353	.1830	.1414	.1151
	SCA-SRNet	**.3014**	**.2159**	**.1644**	**.1290**	**.1026**
WOW	SCA-YeNet	.2442	.1691	.1229	.0959	.0906
	SRNet	.2587	.1676	.1197	.0893	.0672
	SCA-SRNet	**.2197**	**.1401**	**.098**	**.0769**	**.0578**

obtain the same quantity as $\delta_{uSA}^{1/2}$ from [15] (see Eq. (20) in [15] and the discussion following this equation).

The SCA-SRNet was trained in the exact same fashion as the original network. The results for spatial-domain steganography are shown in Table IV. The gain is the largest for WOW as has always been observed in all prior art on SCA steganalysis because WOW is "overly content adaptive". The gain w.r.t. the original SRNet is around 1% for payloads 0.4 and 0.5 bpp and then steadily increases to 4% for the smallest tested payload 0.1 bpp. For HILL and S-UNIWARD, the gain ranges between 1–2%.

The JPEG results appear in Table V. The absolute gain is small for UED-JC for quality factor 75 also because the detection error is already rather small even with the original SRNet. For more difficult cases, such as higher quality factors or smaller payloads, however, the SCA SRNet gains up to 5%, which is rather significant.

Table V
Effect of introducing the selection channel into SRNet (JPEG domain).

Embedding Method	Detector	QF 75 0.1	0.2	0.3	0.4	0.5	QF 95 0.1	0.2	0.3	0.4	0.5
J-UNI	SRNet	.3201	.1889	.1153	.0670	.0385	.4277	.3440	.2516	.1762	.1148
	SCA-SRNet	**.2690**	**.1626**	**.0921**	**.0578**	**.0321**	**.3712**	**.3243**	**.2346**	**.1640**	**.1096**
UED-JC	SRNet	.1311	.0568	.0285	.0188	.0093	.3044	.2028	.1261	.0877	.0500
	SCA-SRNet	**.1245**	**.0524**	**.0265**	**.0153**	**.0084**	**.2774**	**.1672**	**.1068**	**.0663**	**.0395**

VI. Conclusions

A novel convolutional neural network architecture called SRNet is proposed for steganalysis of digital images. SRNet is the first steganalysis network that is free of many externally introduced design elements previously proposed specifically for steganalysis and forensics, such as constrained kernels, initialization with heuristic kernels, thresholding, quantization, and awareness of JPEG phase. Consequently, SRNet can be trained in an end-to-end fashion from randomly initialized convolutional kernels and in the same fashion independently of the embedding domain. The front part of SRNet contains seven residual layers in which pooling has been disabled to allow the network to learn relevant "noise residuals" for different types of embedding changes in both spatial and JPEG domain. The design of SRNet is validated experimentally on standard datasets and six steganographic algorithms. State-of-the-art detection is observed in both domains with rather significant improvements in the JPEG domain. Receiver operating characteristics for selected combinations of embedding algorithms and payloads reveal especially favorable detection performance for very low false-alarm rates, which is expected to be significant for practitioners.

While SRNet was intentionally designed to minimize the use of heuristic design elements specific to steganalysis, it still benefits from being informed about the probabilistic impact of embedding in the form of the selection channel, which points out a space for future improvements. SRNet is the first steganalysis network that makes use of the selection channel for JPEG domain steganalysis, a task that was achieved by adding a bound on L_1 embedding distortion to the feature maps outputted by the first layer to reinforce the output of neurons that are most affected by embedding.

This paper opens up a direction in steganalysis that we plan to further pursue in the future. Since steganalysis detectors by definition detect inconsistencies in the noise patterns of images, they often find applications in forensics, such as for establishing the processing history of images or detecting inconsistencies within a single image to identify locally manipulated regions.

Large advancements in steganalysis need to be followed by revisiting the inner workings of steganographic methods because they are often designed from feedback provided by detectors. A lucrative possibility that has already received attention from researchers [55] is to let two competing networks design the embedding algorithm within the generative-adversarial network (GAN) [24] setup that essentially mimics the game played by the steganographer and the steganalyst. Novel steganalysis architectures, such as the SRNet, will undoubtedly find their place to further advance this direction.

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

References

[1] I. Avcibas, N. D. Memon, and B. Sankur. Steganalysis using image quality metrics. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents III*, volume 4314, pages 523–531, San Jose, CA, January 22–25, 2001.

[2] I. Avcibas, N. D. Memon, and B. Sankur. Image steganalysis with binary similarity measures. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2002*, volume 3, pages 645–648, Rochester, NY, September 22–25, 2002.

[3] 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.

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

[5] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the ICML*, pages 41–48. ACM, 2009.

[6] R. Böhme. *Advanced Statistical Steganalysis*. Springer-Verlag, Berlin Heidelberg, 2010.

[7] M. Boroumand and J. Fridrich. Boosting steganalysis with explicit feature maps. In F. Perez-Gonzales, F. Cayre, and P. Bas, editors, *The 4th ACM Workshop on Information Hiding and Multimedia Security*, Vigo, Spain, June 20–22, 2016.

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

[9] 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.

[10] Mo Chen, V. Sedighi, M. Boroumand, and J. Fridrich. JPEG-phase-aware convolutional neural network for steganalysis of JPEG images. In M. Stamm and M. Kirchner, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '17, Philadelphia, June 20–22, 2017.

[11] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). *CoRR*, 2015. http://arxiv.org/abs/1511.07289.

[12] R. Cogranne and J. Fridrich. Modeling and extending the ensemble classifier for steganalysis of digital images using hypothesis testing theory. *IEEE Transactions on Information Forensics and Security*, 10(2):2627–2642, December 2015.

[13] R. Cogranne, V. Sedighi, T. Pevný, and J. Fridrich. Is ensemble classifier needed for steganalysis in high-dimensional feature spaces? In *IEEE International Workshop on Information Forensics and Security*, Rome, Italy, November 16–19 2015.

[14] 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.

[15] 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.

[16] T. Denemark, J. Fridrich, and P. Comesaña-Alfaro. Improving selection-channel-aware steganalysis features. In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2016*, San Francisco, CA, February 14–18, 2016.

[17] 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.

[18] H. Farid and L. Siwei. Detecting hidden messages using higher-order statistics and support vector machines. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of Lecture Notes in Computer Science, pages 340–354, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[19] 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.

[20] C. Fuji-Tsang and J. Fridrich. Steganalyzing images of arbitrary size with CNNs. In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2018*, Burlingame, CA, January 29–February 2, 2018.

[21] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[22] M. Goljan and J. Fridrich. CFA-aware features for steganalysis of color images. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[23] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*, volume 1. MIT Press Cambridge, 2016.

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[25] 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.

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 27–30 2016.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision ECCV*, volume 9908 of Lecture Notes in Computer Science, Amsterdam, Octoer 8–16 2016.

[28] 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.

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

[30] 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.

[31] 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.

[32] G. Huang, Z. Liu, L. van der Maaten, and K. Q Weinberger. Densely connected convolutional networks. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, July 21–26, 2017.

[33] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[34] 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.

[35] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, 2014. http://arxiv.org/abs/1412.6980.

[36] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. Self-normalizing neural networks. *CoRR*, 2017. http://arxiv.org/abs/1706.02515.

[37] J. Kodovský and J. Fridrich. JPEG-compatibility steganalysis using block-histogram of recompression artifacts. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 78–93, Berkeley, California, May 15–18, 2012.

[38] 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.

[39] J. Kodovský, J. Fridrich, and V. Holub. On dangers of overtraining steganography to incomplete cover model. In J. Dittmann, S. Craver, and C. Heitzenrater, editors, *Proceedings of the 13th ACM Multimedia & Security Workshop*, pages 69–76, Niagara Falls, NY, September 29–30, 2011.

[40] 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.

[41] B. Li, M. Wang, and J. Huang. A new cost function for spatial image steganography. In *Proceedings IEEE, International Conference on Image Processing, ICIP*, Paris, France, October 27–30, 2014.

[42] W. Luo, Y. Wang, and J. Huang. Security analysis on spatial ± 1 steganography for JPEG decompressed images. *IEEE Signal Processing Letters*, 18(1):39–42, 2011.

[43] T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 75–84, Princeton, NJ, September 7–8, 2009.

[44] T. Pevný and A. D. Ker. Towards dependable steganalysis. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, pages 1501–1514, San Francisco, CA, February 8–12, 2015.

[45] 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.

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

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

[48] V. Sedighi and J. Fridrich. Effect of imprecise knowledge of the selection channel on steganalysis. In J. Fridrich, P. Comesana, and A. Alattar, editors, *3rd ACM IH&MMSec. Workshop*, Portland, Oregon, June 17–19, 2015.

[49] Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437

of Lecture Notes in Computer Science, pages 249–264, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.
[50] 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.
[51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.
[52] S. Tan and B. Li. Stacked convolutional auto-encoders for steganalysis of digital images. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pages 1–4, December 2014.
[53] 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.
[54] W. Tang, H. Li, W. Luo, and J. Huang. Adaptive steganalysis based on embedding probabilities of pixels. *IEEE Transactions on Information Forensics and Security*, 11(4):734–745, 2016.
[55] W. Tang, S. Tan, B. Li, and J. Huang. Automatic steganographic distortion learning using a generative adversarial network. *IEEE Signal Processing Letters*, 24(10):1547–1551, October 2017.
[56] T. Thai, R. Cogranne, and F. Retraint. Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm. *IEEE Transactions on Image Processing*, 23(5):1–14, May 2014.
[57] T. H. Thai, R. Cogranne, and F. Retraint. Optimal detection of OutGuess using an accurate model of DCT coefficients. In *Sixth IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, December 3–5, 2014.
[58] S. Wu, S.-H. Zhong, and Y. Liu. Steganalysis via deep residual network. In *IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, Wuhan, China, December 13–16, 2016.
[59] C. Xia, Q. Guan, X. Zhao, Z. Xu, and Y. Ma. Improving GFR steganalysis features by using Gabor symmetry and weighted histograms. In M. Stamm and M. Kirchner, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, June 20–22, 2017.
[60] G. Xu. Deep convolutional neural network to detect J-UNIWARD. In M. Stamm and M. Kirchner, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, June 20–22, 2017.
[61] 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.
[62] 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.
[63] J. Yang, K. Liu, X. Kang, E. Wong, and Y. Shi. Steganalysis based on awareness of selection-channel and deep learning. In *Internartional Workshop on Digital Forensics and Watermarking*, volume 10431 of *LNCS*, pages 263–272, 2017.
[64] J. Yang, Y.-Q. Shi, E.K. Wong, and X. Kang. JPEG steganalysis based on densenet. *CoRR*, abs/1711.09335, 2017.
[65] 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.
[66] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.
[67] J. Zeng, S. Tan, B. Li, and J. Huang. Large-scale JPEG image steganalysis using hybrid deep-learning framework. *IEEE Transactions on Information Forensics and Security*, 13(5):1200–1214, 2018.
[68] D. Zou, Y. Q. Shi, W. Su, and G. Xuan. Steganalysis based on Markov model of thresholded prediction-error image. In *Proceedings IEEE, International Conference on Multimedia and Expo*, pages 1365–1368, Toronto, Canada, July 9–12, 2006.

Mehdi Boroumand received the B.S. degree in electrical engineering from the K. N. Toosi University of Technology, Iran, in 2004 and the M.S. degree in electrical engineering from the Sahand University of Technology, Iran, in 2007. He is currently pursuing the Ph.D. degree in electrical engineering with Binghamton University, State University of New York. His areas of research interest include digital image steganalysis and steganography, digital image forensics, image processing and computer vison, and machine learning.

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 2007, he was a post-doc research associate at the Department of Electrical and Computer Engineering, Binghamton University. From 2009, he has been an adjunct research scientist at SUNY research foundation. Since 2007, he has been working as a principle processing engineer at JADAK LLC, a Novanta Company, responsible for designing the inspecting and tracking machine vision OEM systems for the healthcare automation and vitro diagnostic applications. His research interests include machine vision and machine learning, digital image and video processing, and digital forensics.

Jessica Fridrich holds the position of Distinguished Professor of Electrical and Computer Engineering at Binghamton University (SUNY). She has 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, digital watermarking, and digital image forensics. Dr. Fridrich's research work has been generously supported by the US Air Force, NSF, and AFOSR. Since 1995, she received 23 research grants totaling over $11 mil for projects on data embedding, digital forensics, and steganalysis that lead to 200 papers and 7 US patents. Dr. Fridrich is IEEE Fellow and a member of ACM.