An Intriguing Struggle of CNNs in JPEG Steganalysis and the OneHot Solution

Yassine Yousfi and Jessica Fridrich, *Fellow, IEEE*

***Abstract*—Deep convolutional neural networks (CNNs) have become the tool of choice for steganalysis because they outperform older feature-based detectors by a large margin. However, recent work points at cases where feature-based detectors perform better than CNNs due to their failure to compute simple statistics of DCT coefficients. We introduce a shallow "OneHot" CNN, which encodes DCT coefficients using clipped one-hot encoding into a binary volumetric representation of the DCT plane fed to a convolutional block designed to learn relevant intra-block and inter-block relationships using vanilla and dilated convolutions. Methodology for plugging the "OneHot" network into conventional steganalysis CNNs is also introduced for an end-to-end learnable detector with improved performance.**

***Index Terms*—Steganography, steganalysis, convolutional neural networks, deep learning**

I. Introduction

While CNN detectors [27], [30], [3] clearly outperform classifiers with hand-crafted feature sets for steganalysis in both JPEG and spatial domain (see, e. g., the detailed survey [6]), there is recent evidence that CNNs unexpectedly struggle to perform well in certain cases:

- All ALASKA steganalysis challenge participants [28], [9] consistently underperformed on nsF5 [12].
- In [4], SRNet [3] does not follow the theoretically predicted trend for nsF5 [12], while all other tested steganographic schemes follow the model.
- J-UNIWARD [16] is surprisingly best detected in JPEGs obtained with the "Trunc" quantizer [5] by JPEG rich model (JRM) [18] and not a CNN.

Figure 1 shows the total detection error under equal priors P_{E} for two scenarios in which two leading CNN architectures for JPEG domain steganalysis, the SRNet and J-XuNet [27], are outperformed by an older detection paradigm, the JRM model and the ensemble classifier [19]. In this paper, we analyze these intriguing failures and address the deficiency with a shallow CNN, the "OneHot" CNN, that can be plugged into a conventional CNN architecture as a dual branch for an end-to-end learnable detector.

In Section II, we study SRNet and its variants on nsF5, and link its struggles to the inability to "see" simple artifacts in the distribution of DCT coefficients exploited by the JRM. After briefly reviewing prior art on CNNs with DCT inputs, in Section III we introduce a shallow "OneHot" CNN that can be plugged to SRNet and trained in an end-to-end fashion to address the above struggles (Section IV). The paper is concluded in Section V. Datasets (JPEG round/trunc sources), performance measures, and some technical aspects of training are detailed in Section VI.

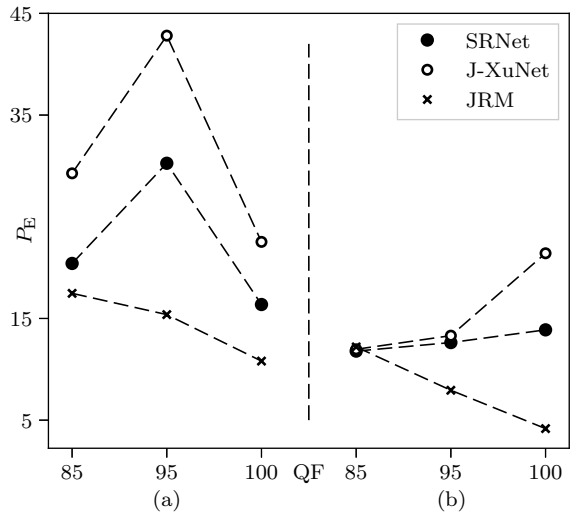

Figure 1. Detection error P_{E} of SRNet, J-XuNet, and JRM+ensemble for (a) J-UNIWARD 0.4 bpnzac in JPEG trunc source and (b) nsF5 0.2 bpnzac in JPEG round source.

II. Struggles of CNNs in JPEG steganalysis

Figure 1 shows that there exist cases where CNNs underperform by a large margin when compared to JRM, which is a rather simple feature set. Examining each JRM submodel (Figure 2) separately reveals that most of the detection performance is due to the sub-model 'Ax_T5' corresponding to integral co-occurences from absolute values of the DCT plane computed with a clipping threshold $T = 5$. CNNs fed with decompressed JPEGs are apparently unable to see artifacts in the distribution of DCTs, such as the co-occurrence 'Ax_T5'.

Feeding the array of DCTs directly to SRNet, however, failed to produce reliable detection or did not converge (DNC). We hypothesize that this is due to the fact that, unlike pixels, DCTs are largely decorrelated and locally heterogeneous, making it harder for the convolutions to extract relevant image components and noise statistics. Most effort in computer vision directed towards training on DCT inputs has focused on either avoiding the costly JPEG decompression step to speed-up training [21] or on approximating a CNN trained on spatial-

This material is based on research sponsored by 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 thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

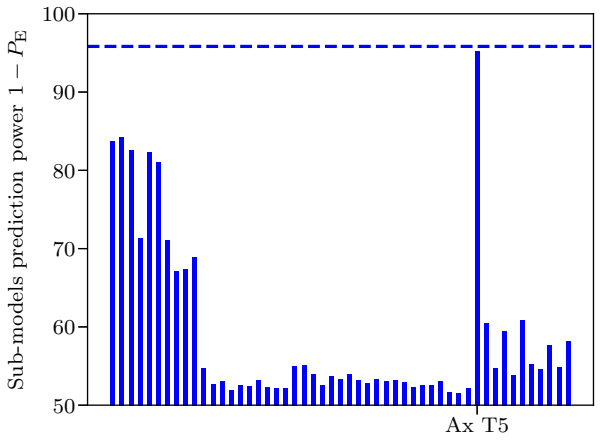

Figure 2. $1-P_E$ when staganalyzing nsF5 0.2 bpnzac in JPEG round QF100 using individual JRM submodels and using the entire JRM (dashed line).

Table I
DETECTION ERROR OF SRNET AND ITS TWO SHALLOWER VARIANTS USING THE ORIGINAL AND LONGER TRAINING SCHEDULE FOR NSF5 0.2 BPNZAC JPEG ROUND QF 100.

Architecture description	Original schedule	Longer schedule
SRNet	DNC	5.35
SRNet, Layers1-8+Global Average Pooling+FC (fully connected)	13.36	9.66
SRNet, Layers8-12+FC	25.46	19.84
JRM	**4.17**	

domain inputs [10]. Neither is relevant for our needs. In [24], a histogram layer is introduced that can compute predefined higher-order statistics, which would merely mimic the JRM.

In our case, we found out that adjusting the training schedule partially solved the problem with convergence and loss of performance at the expense of a *significantly* longer training time. Table I shows the results of SRNet with DCT inputs for nsF5 0.2 bpnzac in JPEG round QF 100 using the original training schedule [3] and a longer schedule using a doubled batch-size of 64 and seeding from a much larger payload 0.4 bpnzac. We also studied shallower versions of SRNet by pruning different layers to show that this difficulty is not linked to an excessive number of parameters to learn. Note that when using the Titan Xp GPU, SRNet's longer schedule takes 5 days to train compared to 1.2 days for the original training schedule.

III. ONEHOT CNN

A simple transformation of the input DCT plane (the "clipped one-hot encoding") before the first convolution will allow a CNN compute occurrences and co-occurrence histograms. The DCT array $\mathbf{M}$ is first clipped to a threshold T and then transformed to a binary volume of size $(T+1)\times H\times W$:

$$\begin{aligned}\mathbb{Z}^{H\times W} &\to \{0,1\}^{(T+1)\times H\times W}\\ \mathbf{M} &\mapsto \left\{\begin{array}{ll}[|\mathbf{M}|=t], & t\in\{0,...,T-1\}\\ [|\mathbf{M}|\geq t], & t=T\end{array}\right\}\end{aligned} \tag{1}$$

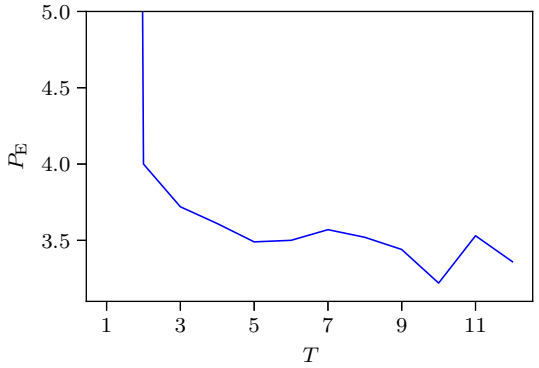

Figure 3. P_E of OneHot CNN on nsF5 0.2 bpnzac in round QF 100 JPEGs for different clipping thresholds T.

where $[.]$ is the (element-wise) Iverson bracket $[.]$. In fact, one can even find a specific kernel that will compute a desired histogram. For example, horizontal co-occurrences for coefficient pairs $(x,y)\in\{0,...,T\}^2$ can be computed by convolving the input volume with the following convolutional kernel $\mathbf{K}\in\mathbb{R}^{(T+1)\times 3\times 3}$, followed by global average pooling

$$\mathbf{K}=\begin{cases}\mathbf{K}_t=\left(\begin{smallmatrix}0&0&0\\0&1&0\\0&0&0\end{smallmatrix}\right), & t=x\\ \mathbf{K}_t=\left(\begin{smallmatrix}0&0&0\\0&0&1\\0&0&0\end{smallmatrix}\right), & t=y\\ \mathbf{K}_t=0_{\mathbb{R}^{3\times 3}} & \text{else.}\end{cases} \tag{2}$$

Inter-block statistics can be designed similarly by using dilated convolutions with rate 8 introduced in wavelet decompositions algorithms [15], also referred to as "à trous" convolutions widely used in computer vision [7], [23], [29] as a way to increase the receptive field of convolutional layers.

Figure 4 shows the overall architecture of the proposed OneHot network. Note that the "clipping" operation is necessary for memory constraints. Figure 3 shows how the detection error P_E reacts to different clipping thresholds. While $T=10$ seems to be optimal, in practice any $T\geq 5$ can be chosen, as improvements recorded for higher thresholds are less than 0.5%. Tuning D_1 and D_2 also seems to have a rather minor effect on performance. However, setting $(D_1,D_1')=(64,0)$ or $(0,64)$, i. e., using only dilated convolutions or only vanilla convolutions, respectively, seems to hurt the detection performance by more than 2%. Table II shows that the proposed OneHot CNN performs better than JRM in the two problematic cases introduced in Section I.

The OneHot CNN is trained with the same training schedule and hyper-parameters as SRNet, and takes around 13 hours on NVIDIA's Titan Xp GPU.

A. Cartesian Calibration

Cartesian calibration [13], [17] is a way to augment any feature set by adopting additional features computed from a reference image. The reference image is obtained by decompressing the original image, cropping by four pixels in both directions, and recompressing the cropped image. It has been

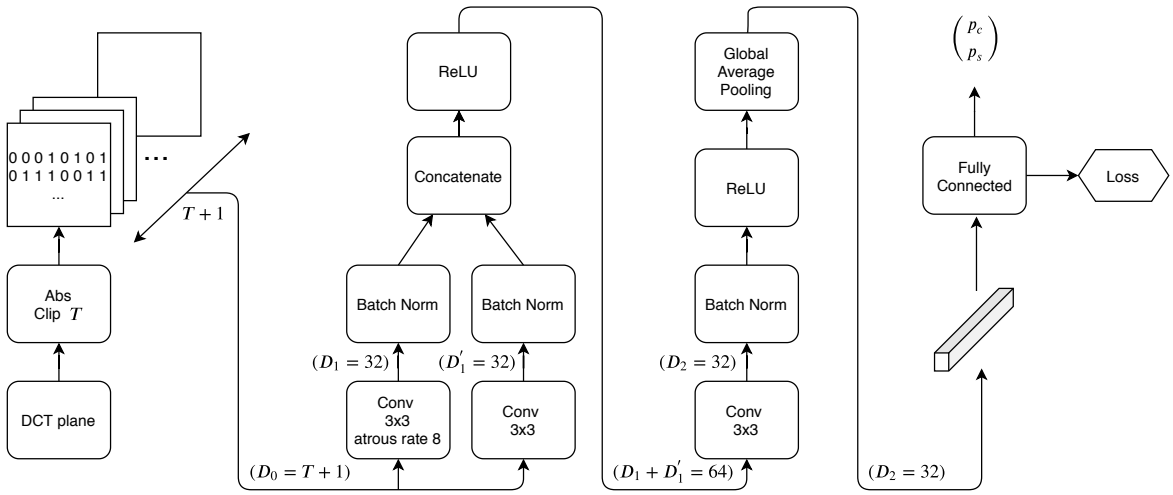

Figure 4. OneHot CNN architecture. D_i corresponds to the depth of representations at different layers of the network.

Table II
DETECTION ERROR P_{E} OF JRM AND THE ONEHOT CNN FOR THE TWO PROBLEMATIC CASES: JPEG ROUND, NSF5 AND JPEG TRUNC, J-UNIWARD FOR VARIOUS QUALITY FACTORS AND PAYLOADS.

QF	100		95		85	
Round, nsF5	0.2	0.1	0.2	0.1	0.2	0.1
JRM	4.17	21.99	7.94	27.37	12.2	30.49
OneHot CNN	**3.49**	**20.65**	**7.90**	**27.06**	**11.28**	**29.95**
Trunc, J-UNI	0.4	0.3	0.4	0.3	0.4	0.3
JRM	10.81	19.60	15.38	23.18	17.47	24.49
OneHot CNN	**7.36**	**14.05**	**14.32**	**21.55**	**16.45**	**22.79**

Table III
DETECTION ERROR P_{E} OF CCJRM AND THE CALIBRATED ONEHOT CNN FOR JPEG ROUND, NSF5 FOR VARIOUS QUALITY FACTORS AND PAYLOADS.

QF	100		95		85	
Round, nsF5	0.2	0.1	0.2	0.1	0.2	0.1
ccJRM	2.11	18.03	7.15	26.81	10.80	29.78
ccOneHot CNN	**1.39**	**15.52**	**6.66**	**25.75**	**10.09**	**29.26**

shown that cartesian calibration helps nsF5, Jsteg, YASS, and other steganographic schemes [17].

We show that the OneHot CNN can also be augmented with cartesian calibration by adding a second OneHot branch taking the reference image as input. Both branches are independent until the fully FC layer, which takes a concatenation of the 2×32 representation as inputs. Denoting this architecture ccOneHot CNN, Table III shows its superior performance w.r.t. ccJRM.

IV. ONEHOT+SRNET

In this section, we show that merging the OneHot network with conventional CNN architectures produces more universal detectors. We use SRNet to show how these two networks are merged and how the resulting architecture denoted OneHot+SRNet compares to simply concatenating (a trained) SRNet's last layer and JRM features as a feature set and training FLD ensemble. This strategy is denoted JRM+SRNet.

The OneHot+SRNet is built by merging SRNet and OneHot in a branch-parallel fashion, each branch B outputs a feature extraction $\mathcal{F}_B$ (the output of the layer before FC_B, the fully connected layer of branch B) and a binary output $\mathbf{p}_B$ (the output of the FC_B+softmax). $\mathcal{F}_{SRNet}$ and $\mathcal{F}_{OneHot}$ are then concatenated and fed to the final FC layer, which outputs $\mathbf{p}_{FC}$, the final classification probability.

For each component B (SRNet, OneHot, and FC), we use the binary cross-entropy loss: $\mathcal{L}_B = -(y \log(p_B^{\text{s}}) + (1 - y) \log(p_B^{\text{c}}))$, where y is the binary target, and combine the losses as follows:

$$\mathcal{L} = \sum_{B \in \{SRNet,\, OneHot,\, FC\}} \lambda_B \mathcal{L}_B, \tag{3}$$

where λ_B is a scaling hyperparameter for each branch B weighting the importance of training the branch B compared to other branches. The weights can be assigned manually as done in [11], [20], [25], or heuristically modeled as noise parameters and learned as done in [8]. In the following experiments we set all $\lambda_B = 1$.

Another key element in this merging architecture is making sure that each weight in the network is only updated once during back-propagation, which is done by "stopping" the gradients at the input of the merged FC layer, to ensure that the gradients of $\mathcal{L}_{FC}$ are not computed w.r.t. to the weights of SRNet and OneHot.

Table IV shows that this strategy works well in practice. The first two blocks of the table show that OneHot+SRNet substantially improves SRNet. For nsF5 in JPEG round, the improvement is comparable to JRM+SRNet. For J-UNIWARD in JPEG trunc, the improvement is consistently better than JRM+SRNet. The next two blocks show that OneHot+SRNet has comparable detection performance to SRNet for J-UNIWARD and UED-JC [14] in JPEG round while avoiding some large degradations of JRM+SRNet (e. g., J-UNIWARD for QF 100 and 95).

In Table V, we show that OneHot+SRNet substantially improves detection in a more diverse dataset. We use ALASKA

Table IV
DETECTION ERROR P_{E} FOR VARIOUS JPEG QUALITY FACTORS, ROUND/TRUNC, EMBEDDING SCHEMES, AND PAYLOADS.

QF	100		95		85	
Round, nsF5	0.2	0.1	0.2	0.1	0.2	0.1
SRNet	13.88	30.76	12.63	25.35	11.70	24.39
JRM+SRNet	**1.97**	**18.11**	3.50	**19.39**	3.63	19.42
OneHot+SRNet	1.99	18.55	**3.32**	19.73	**3.50**	**19.22**
Trunc, J-UNI	0.4	0.3	0.4	0.3	0.4	0.3
SRNet	16.37	21.26	30.26	35.82	20.41	26.26
JRM+SRNet	7.62	17.49	14.32	22.87	10.14	17.76
OneHot+SRNet	**7.29**	**13.64**	**14.18**	**21.79**	**10.13**	**16.16**
Round, J-UNI	0.4	0.3	0.4	0.3	0.4	0.3
SRNet	12.52	**16.70**	**17.40**	**24.39**	9.17	14.32
JRM+SRNet	15.32	18.64	17.94	26.74	9.18	14.44
OneHot+SRNet	**11.94**	16.99	17.52	24.81	**8.84**	**14.04**
Round, UED-JC	0.3	0.2	0.3	0.2	0.3	0.2
SRNet	**6.96**	**10.16**	**10.90**	**17.56**	4.44	**7.26**
JRM+SRNet	7.69	10.53	10.99	17.86	**4.04**	7.38
OneHot+SRNet	7.26	10.58	11.35	18.25	4.40	7.58

Table V
DETECTION ERROR P_{E} AND MISSED DETECTION RATE AT 5% FALSE ALARM, $MD5$, FOR ALASKA V1 WHEN TESTED AGAINST INDIVIDUAL STEGO SCHEMES AND THEIR MIXTURE.

ALASKA v1 QF95	SRNet	OneHot+SRNet
J-UNI	10.63, 18.34	**10.97, 18.20**
EBS	**8.21, 11.51**	8.24, 11.71
UED	**10.97, 17.97**	12.04, 20.68
nsF5	27.90, 70.86	**16.37, 34.02**
Mixture	12.96, 25.08	**12.01, 20.34**

v1 256×256 tiles as described in VI compressed with JPEG quality factor 95. As prescribed by the challenge winners [28], the detectors are trained as multi-class and used as binary detectors. Color channels for SRNet are merged in the first layer by using $3 \times 3 \times 3$ convolution kernels, and the "clipped one-hot encoding" in the OneHot branch is done separately for each channel using the same threshold, producing a volume of size $3(T+1) \times H \times W$. Note that for ALASKA v1 we use $\lambda_{SRNet} = 4$ and $\lambda_{FC} = \lambda_{OneHot} = 1$ as it gave the best results.

V. DISCUSSION AND CONCLUSIONS

While in other fields CNNs have been reported to be underperforming, for example, in solving the seemingly trivial coordinate transform problem [22], to the best of the authors' knowledge, no prior art uncovered failings of CNNs in steganalysis. In this work, a new CNN architecture is proposed, the OneHot CNN, to overcome struggles of CNNs in at least two particular scenarios reported in this paper. It is based on the clipped one-hot encoding, which enables computing higher-order statistics of DCT coefficients in a flexible learnable manner.

The paper additionally describes a dual-branch architecture for adding the OneHot CNN to existing CNNs for steganalysis (SRNet) for an end-to-end trainable detector. This overcomes the reported struggles while not decreasing the performance in cases when the OneHot branch is not effective. For ALASKA v1 and QF 95, OneHot+SRNet tile detector performs 4.7% better than SRNet in terms of $MD5$ by substantially improving the detection of nsF5.

We anticipate that the proposed OneHot architecture will find applications in forensics for detection of higher-order artifacts in the distribution of DCT coefficients. All code used to produce the results in this paper, including the network configuration files, will be made available from http://dde.binghamton.edu/download/ upon acceptance of this paper.

VI. SETUP OF EXPERIMENTS

Unless mentioned otherwise, all experiments were executed on the union of BOSSbase 1.01 [1] and BOWS2 [2] converted to grayscale and resized to 256×256 using Matlab's 'imresize' with default parameters. As in [3], [4], the dataset was randomly divided into three sets with 14,000 (BOSSbase+BOWS2) / 1,000 (BOSSbase) / 5,000 images (BOSSbase) for training, validation, and testing. The "trunc" and "round" sources correspond to images where the final DCT quantizer in JPEG compression is truncation towards zero and round, respectively.

In Section IV, ALASKA v1 [9] dataset has been used with the scripts adapted to produce 256×256 crops with JPEG compression only done in the "round" mode. This dataset was randomly divided into three sets with 42,500 / 3,500 / 3,500 for training, validation, and testing. The splits were made to be compatible with the datasets used in [28].

The steganographic algorithms used in this paper are: J-UNIWARD [16], UED-JC [14], EBS [26], and nsF5 [12], embedded with fixed payloads (BOSS+BOWS2) or adaptive payload based on the image processing history, with priors 0.4, 0.3, 0.15, and 0.15, respectively (ALASKA v1). In ALASKA v1, color steganography is done by spreading the payload between Y, C_r, and C_b as described in [9] (*Payload repartition among color channels*).

A. Data augmentation in DCT domain

The first step to using DCT domain inputs in deep learning is to perform data augmentation in the DCT domain. Rotations by multiples of $\pi/2$ and horizontal/vertical flips can be done in a lossless fashion directly on DCT coefficients $\mathbf{M} \in \mathbb{Z}^{H \times W}$ thanks to the symmetries of DCT bases. We denote $f_8(\mathbf{M})$ any operation f performed in a block-wise fashion, with a block size of 8, e. g., T_8 is the 8×8 block-wise matrix transpose. For simplicity, we introduce $J = [(-1)^j]_{\substack{0 \le i < H \\ 0 \le j < W}}$ and $I = [(-1)^i]_{\substack{0 \le i < H \\ 0 \le j < W}} \in \{1, -1\}^{H \times W}$, all-ones matrices with a negative sign in odd columns and rows, respectively. Eqs. 4 and 5 show how to vertically flip and rotate by $\pi/2$

$$\text{Lossless flip}^V(\mathbf{M}) = J \cdot \text{flip}_8^V \circ \text{flip}^V(\mathbf{M}) \tag{4}$$

$$\text{Lossless rot}^{\pi/2}(\mathbf{M}) = I \cdot \mathrm{T}_8 \circ \text{rot}_8^{3\pi/2} \circ \text{rot}^{\pi/2}(\mathbf{M}), \tag{5}$$

where $\cdot$ is an element-wise multiplication, and $\circ$ is the composition operation. All other valid flips and rotations can be derived in a similar fashion (or as compositions of 4 and 5).

REFERENCES

[1] P. Bas, T. Filler, and T. Pevný. Break our steganographic system – the ins and outs of organizing BOSS. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, volume 6958 of Lecture Notes in Computer Science, pages 59–70, Prague, Czech Republic, May 18–20, 2011.

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

[3] M. Boroumand, M. Chen, and J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, May 2019.

[4] J. Butora and J. Fridrich. Effect of JPEG quality on steganographic security. In R. Cogranne and L. Verdoliva, editors, *The 7th ACM Workshop on Information Hiding and Multimedia Security*, Paris, France, July 3–5, 2019. ACM Press.

[5] J. Butora and J. Fridrich. Steganography and its detection in JPEG images obtained with the "trunc" quantizer. In *IEEE ICASSP*, Barcelona, Spain, May 4–8, 2020. To appear.

[6] M. Chaumont. Deep learning in steganography and steganalysis from 2015 to 2018. In *Digital Media Steganography: Principles, Algorithms, Advances*, volume abs/1904.01444. Elsevier, 2020.

[7] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[8] R. Cipolla, Y. Gal, and A. Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018.

[9] R. Cogranne, Q. Giboulot, and P. Bas. The ALASKA steganalysis challenge: A first step towards steganalysis âinto the wildâ. In R. Cogranne and L. Verdoliva, editors, *The 7th ACM Workshop on Information Hiding and Multimedia Security*, Paris, France, July 3–5, 2019. ACM Press.

[10] M. Ehrlich and L. S. Davis. Deep residual learning in the JPEG transform domain. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3484–3493, 2019.

[11] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *2015 IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[12] J. Fridrich. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of Lecture Notes in Computer Science, pages 67–81, Toronto, Canada, May 23–25, 2004. Springer-Verlag, New York.

[13] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In *Information Hiding, 5th International Workshop*, volume 2578 of Lecture Notes in Computer Science, pages 310–323, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[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] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. *Wavelets, Time-Frequency Methods and Phase Space*, January 1989.

[16] 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.

[17] J. Kodovský and J. Fridrich. Calibration revisited. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 63–74, Princeton, NJ, September 7–8, 2009.

[18] 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.

[19] 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, April 2012.

[20] I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[21] B. Kadlec R. Liu L. Gueguen, A. Sergeev and J. Yosinski. Faster neural networks straight from JPEG. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3933–3944. Curran Associates, Inc., 2018.

[22] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2018.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[24] V. Sedighi and J. Fridrich. Histogram layer, moving convolutional neural networks towards feature-based steganalysis. In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2017*, San Francisco, CA, January 29–February 1, 2017.

[25] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *2nd International Conference on Learning Representations, (ICLR) 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, April 2014.

[26] C. Wang and J. Ni. An efficient JPEG steganographic scheme based on the block–entropy of DCT coefficents. In *Proc. of IEEE ICASSP*, Kyoto, Japan, March 25–30, 2012.

[27] 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.

[28] Y. Yousfi, J. Fridrich, J. Butora, and Q. Giboulot. Breaking ALASKA: Color separation for steganalysis in JPEG domain. In R. Cogranne and L. Verdoliva, editors, *The 7th ACM Workshop on Information Hiding and Multimedia Security*, Paris, France, July 3–5, 2019. ACM Press.

[29] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[30] 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.