ImageNet Pre-trained CNNs for JPEG Steganalysis

Yassine Yousfi
Department of ECE
Binghamton University
yyousfi1@binghamton.edu

Jan Butora
Department of ECE
Binghamton University
jbutora1@binghamton.edu

Eugene Khvedchenya
ODS.ai
eekhvedchenya@gmail.com

Jessica Fridrich
Department of ECE
Binghamton University
fridrich@binghamton.edu

***Abstract*—In this paper, we investigate pre-trained computer-vision deep architectures, such as the EfficientNet, MixNet, and ResNet for steganalysis. These models pre-trained on ImageNet can be rather quickly refined for JPEG steganalysis while offering significantly better performance than CNNs designed purposely for steganalysis, such as the SRNet, trained from scratch. We show how different architectures compare on the ALASKA II dataset. We demonstrate that avoiding pooling/stride in the first layers enables better performance, as noticed by other top competitors, which aligns with the design choices of many CNNs designed for steganalysis. We also show how pre-trained computer-vision deep architectures perform on the ALASKA I dataset.**

***Index Terms*—Steganography, steganalysis, convolutional neural network, pre-trained models, EfficientNet, MixNet, SRNet, transfer learning**

I. Introduction

Steganography is a form of covert communication in which messages are hidden inside other objects overtly communicated to the recipient. The objective of steganalysis is to establish the use of steganography. As more advanced steganographic techniques appeared, steganalysts turned to machine learning to train detectors to recognize subtle but anomalous higher-order statistics left behind by steganography. This was initially executed by representing images with "features" (vectors) hand-designed [1], [2] to be sensitive to steganographic embedding changes and then training a classifier to recognize the discrepancy between the features of cover and stego images. Linear classifiers [3], support-vector machines [4], and ensembles of simple detectors [5] were proposed to handle this task. This approach culminated in what is recognized today as the so-called rich media models [6], [7], [8].

Five years ago, this well-established approach began being replaced by an even more automatized process based on convolutional neural networks (CNNs). The images themselves, rather than their representations, are fed to the network, which learns how to internally represent and classify them via a parametrized hierarchical structure that can be optimized with an efficient gradient-based optimization algorithm. Today, CNNs firmly established themselves as far more powerful then classifiers trained on rich representations. Initially, either fixed high-pass filters or learnable filters but pre-initialized to those from the Spatial Rich Model [6] or to DCT bases have been used in the first convolutional layer [9], [10], [11], [12]. This element was deemed as essential and beneficial for the network to converge and perform well. Researchers experimented with different activation functions, pre-processing modules [13], [14], unpooled layers [11], [15], depth-wise separable convolutions [16], multi-level pooling [17], residual shortcut connections [15], [18], dense connections [19], hybrid designs [20], parallel subnets [13], [20], wider separable architectures [16], while some introduced specialized steganalysis concepts into CNNs, such as the JPEG phase [14] or the reference channel [21].

In this paper, we share our experience with recent computer-vision models originally pre-trained on ImageNet [22] for image classification, which were refined for steganalysis in the JPEG domain. This approach was predominantly employed by virtually all top performers during the recent steganalysis competition ALASKA II hosted on Kaggle. While our exposition is inevitably limited in many ways due to the very short time between the competition end and the paper submission deadline, we strongly believe that the insight we obtained will be valuable to practitioners and researchers. Pre-training exposes the CNN to more than a million images from a very large number of sources, extremely diverse processing, and diverse content. As such, the filters in their convolutional layers are able to recognize a great diversity of shapes, textures, noise patterns, processing and image-development traces, which are exactly the attributes that modulate the stego signal of modern content-adaptive steganographic schemes. Detecting stego noise is essentially equivalent to detecting traces of the content itself.

This approach is fairly new to the steganalysis literature. In fact, to the best of the authors knowledge, only one published paper [23] uses an ImageNet pre-trained model for steganalysis in spatial domain but does not compare it to any other steganalysis detector, nor uses a standard dataset.

In the next section, we introduce ImageNet models and methods of transfer learning for steganalysis. Section III describes the experimental setup, Section IV-A lays out the paper's main experimental results, Section V briefly describes the ALASKA II competition and our prize winning submission. The paper is concluded in Section VI.

II. ImageNet models

ImageNet is one of the largest computer vision benchmark databases. Most computer vision research uses the Ima-

WIFS'2020, December, 6-11, 2020, New York, USA. 978-1-7281-9930-6/20/$31.00 *©2020 IEEE.*

geNet Large Scale Visual Recognition Challenge (ILSVRC) "trimmed" version of 1,000 classes and approximately 1.3 million training images.

This paper experiments with CNNs originally trained on ImgeNet and with a model scaling parameter that controls the size of the network: ResNet [24] and its variants, TResNet [25], SK-ResNeXt [26], and DenseNet [27], as well as the EfficientNet [28] and MixNet [29] (MN).

In addition to recent advances in neural architectures benchmarked on ImageNet, a large body of work is dedicated to transfer learning, i. e., using ImageNet pre-trained neural networks and fine-tuning either the entire network or a subset of the network on a new task. A recent study of transfer learning of ImageNet models [30] shows that better ImageNet models transfer better to new tasks.

A. Steganalysis transfer learning

Steganalysis transfer learning was done using two pipelines: (A) A pipeline inspired by Alex Shonenkov's public baseline,[1] using cross-entropy loss and the AdamW optimizer with 10^{-2} weight decay, for 50 epochs using "ReduceLROnPlateau" Learning Rate (LR) scheduler monitoring the validation loss with a start LR of 10^{-3}, a patience of 2 epochs, a multiplier of $1/2$, and D4 training augmentation. (B) A slightly different pipeline using a multi-head classifier (Binary and Multi-class heads) and cosine annealing LR decay from 10^{-3} to 10^{-5}, for 100 epochs, and using coarse dropout with a small probability and a maximum number of zeroed regions set to 1, in addition to D4 augmentations. All training augmentations were performed using the Albumentations library [31].

Targeted experiments indicated that the most influential hyper-parameters in fine-tuning ImageNet models using both pipelines are the LR and the weight decay, which have to be adjusted for each optimizer. Due to time constraints, we did not perform a rigorous search for optimal settings but early experiments showed that (A) and (B) worked rather well for various CNNs although (B) gave slightly better results for larger architectures.

Networks larger than 10M parameters were trained using Automatic Mixed Precision (AMP), from NVIDIA's apex library.

Note that the refining detailed here has been used on the ALASKA II training dataset described in Section III. It is possible that it may need to be adjusted for best results when refining other pre-trained models on other datasets, stego methods, and a different set of JPEG quality factors.

III. Experimental setup

The majority of experiments reported upon in this paper were executed on the training dataset made available by the organizers of ALASKA II. It contains 3×25,000 different cover images compressed with quality factors 75, 90, and 95, and the same amount of stego images embedded with J-UNIWARD [32], J-MiPOD [33], and UERD [34], making the training set size 4×75,000 images. Per the description of the organizers, the payload embedded in each image was scaled so that all images are approximately equally difficult to detect, with comparatively smaller payload embedded in smooth images and larger payloads embedded in highly textured or noisy images. The average payload embedded across the database was 0.4 bits per non-zero AC DCT coefficient (bpnzac).

Unless mentioned otherwise, for the purpose of the competition the training set was randomly split into three disjoint subsets while making sure each cover image was in the same subset as its three stego versions: the training, validation, and testing sets with 4×3×22,000, 4×3×1,000, and 4×3×2,000 images, respectively.

Most experiments were evaluated with a performance measure derived from the receiver operating characteristic curve (ROC) defined as the probability of correct detection of stego image as a function of the probability of false alarm, $P_{\rm D}(P_{\rm FA})$: the weighted area under the ROC (wAUC) used in ALASKA II and defined as

$$\text{wAUC} = (2A_{0.4} + A_1)/3, \tag{1}$$

where $A_{0.4}$ is the area between the ROC, the two horizontal lines defined by $P_{\rm D} = 0$ and $P_{\rm D} = 0.4$ and the vertical line $P_{\rm FA} = 1$, and A_1 is the area between the ROC, the horizontal line defined by $P_{\rm D} = 0.4$ and the vertical line $P_{\rm FA} = 1$. Experiments in Section IV-D are evaluated using the missed detection rate MD_5 at false alarm 0.05, $MD_5 = 1 - P_{\rm D}(0.05)$, and the minimum average total error under equal priors $P_{\rm E} = \min \frac{1}{2}(P_{\rm D}(P_{\rm FA}) + P_{\rm FA})$, for consistency with the ALASKA I competition results.

IV. Results

A. Baseline

As a baseline for the current state of the art in steganalysis of JPEG images, we selected the three-channel SRNet [15] ($YCrCb$), an architecture specifically designed for steganalysis. It was trained from randomly initialized weights on QF75 using standard hyper-parameters and the training schedule described in [15] with a batch size of 64. Then, it was fine-tuned on QF90 and QF95 separately for $160,000$ iterations with LR 10^{-4} for the first $100,000$ iterations, which was then divided by 2 after each $20,000$ iterations.

Another version of SRNet has also been studied in an attempt to improve the performance: refining the trained SRNet on QF75 without the cover–stego pair constraint (PC). This was done by training on each QF for $200,000$ iterations with LR 10^{-4} for $20,000$ iterations, 10^{-3} for $60,000$ iterations, and for additional $3 \times 40,000$ iterations after dividing the LR by $10, 5, 2$.

The wAUC for both SRNet versions is shown in Table I broken up by quality factors and stego methods. The improvement due to refinements is especially significant for the two larger quality factors and for UERD and J-UNIWARD. We note that all SRNet versions were trained on non-rounded $YCrCb$ values after decompressing the JPEG image. We strongly hypothesize that the improvement due to refining without pair

[1] https://www.kaggle.com/shonenkov/train-inference-gpu-baseline

constraint is due to restoring batch independence together with using Batch Normalization layers. Pair constraint however, helps convergence early in training from scratch. Also note that SRNet trained on all three quality factors together underperformed compared to dedicated SRNet's trained on each QF as the range is larger compared to those studied in [35].

B. ImageNet models

In Table II, we show the wAUC for five different pre-trained models also broken up by quality factor and stego method. All models were refined as explained in Section II-A, EfficientNet B7* was trained using pipeline (B) while the other models were trained using pipeline (A). Another round of refinement was performed with the Mish activation function [36] replacing the original Swish activation function [37]. The boost in wAUC provided by training on non-rounded pixel values consistently ranged between $0.05-0.1$, while the boost due to Mish activation is visualized in Figure 1. Unlike SRNet, the pre-trained models trained on all three QFs at once performed similarly as models dedicated to a specific QF.

The pre-trained models offer markedly better performance than the SRNet on all quality factors and all stego methods. Also, deeper models generally achieve better detection accuracy. Figure 1 shows the performance in terms of the wAUC as a function of the model size (the number of parameters) across all stego methods and then separately for each stego algorithm. The graphs confirm that "bigger and deeper" is generally better. When viewing the performance for each stego algorithm, however, one can see that the models' accuracy varies quite a bit. For UERD, MixNet-xL reaches basically the same performance as the much bigger B6 or B7*. For J-UNIWARD and J-MiPOD, the SRNet no PC has a competitive performance and even outperforms MixNet-S and B2. The boost due to the Mish activation is mostly for J-UNIWARD and J-MiPOD. This complementary performance of the models is good as they will likely boost each other in an ensemble.

C. Pooling/stride ablation

Targeted experiments using different CNN architectures show that the resolution of the first layers is important for the final accuracy. In fact, it is well established within the steganalysis community that CNNs for steganalysis should not perform any downsampling in the first few layers [11], [15]. Figure 2 shows how models building on the ResNet stem (conv 7×7 with stride 2 followed by a 3×3 max pooling layer with stride 2) compare to the SRNet noPC as a baseline. Architectures with too much downsampling in the first layers do not follow the trend in Figure 1, and are generally weaker than the baseline. Figure 2 also shows two different trends, DenseNet and (SK-)ResNeXt families seem to perform better than ResNet and T-ResNet families. We hypothesize that this is due to the fact that DenseNet-121 and (SK-)ResNeXt-50 have a depth of 256 at the output of the first block after the stem (second highest resolution) while ResNet-34 and T-ResNet-M have a depth of 64.

Next, we compare within a single architecture (MixNet-S) how stride and max pooling in the stem affect the performance. Table III shows the great benefit of removing the stem's stride. Table III also shows how MixNet-S performs with different stem downsampling settings when removing the stride and adding an avg. pooling layer. The performance drops considerably despite the same "resolution" as the vanilla MixNet-S stem. The drop is due to the low-pass nature of the average pooling layer, which suppresses high frequency components. The performance drops even more when keeping the strided convolution.

D. Selected case results: ALASKA I

In addition to ALASKA II experiments, we show how an ImageNet pre-trained model, EfficientNet B4 with Mish activation performs on the ALASKA I dataset [38] with nsF5, UED, EBS, and J-UNIWARD as stego schemes. The dataset preparation scripts have been modified to produce 256×256 images (tiles), embedded with double the original payload size, and compressed with JPEG quality 95. Recent work shows that CNNs trained in the spatial domain struggle to detect nsF5 for high JPEG quality factors [39]. Table IV shows that the EfficientNet family also struggles with nsF5. For the other stego schemes, EfficientNet B4 (Mish) surprisingly performs similar to SRNet. We hypothesize that ImageNet pre-trained models are more data efficient than SRNet trained from scratch – ALASKA I has twice as many images per JPEG quality factor ($25,000$ for ALASKA II and $50,000$ for ALASKA I). With the right design, ImageNet pre-trained models are able to get more reliable performance with less data, which seems to be in line with the observation made in [30]: "4.7. Accuracy benefits of ImageNet pre-training fade quickly with dataset size." Note that EfficientNet B4 (Mish) was trained using pipeline (A) described in Section II-A and initialized with ALASKA II weights. Searching for better hyper-parameters for the ALASKA I dataset might give slightly better results.

V. The ALASKA II Challenge

ALASKA II competitors were evaluated using the wAUC (1) on 5,000 images split into 1,000 from a public and 4,000 from a private leader board (LB). The actual details about the split were unavailable to the teams. Each team was allowed five submissions per day consisting of a scoring of all 5,000 images, with higher score given to images more likely to be stego. The feedback about the detection accuracy was in the form of the wAUC computed only from the 1,000 images from the public LB.

The only information about the test set images that was provided was that each embedding algorithm was used with the same probability and the payload computed in the same fashion as for the training set with the average message length of 0.4 bpnzac. The images were all compressed with one of the three JPEG quality factors: 95, 90, and 75.

Model ensembles, 2nd level stacking, and final submission: Due to the competition's time constraint, and the team's late merger with Eugene Khvedchenya, our model ensemble

Model	UERD			J-UNIWARD			J-MiPOD			Mixture
QF	75	90	95	75	90	95	75	90	95	
SRNet	0.9208	0.9081	0.8987	0.8675	0.8459	0.8499	0.9760	0.9604	0.8199	0.8934
SRNet noPC	**0.9385**	**0.9526**	**0.9391**	**0.8788**	**0.8841**	**0.8851**	**0.9814**	**0.9776**	**0.8501**	**0.9227**

Table I
WAUC FOR SRNET TRAINED WITH COVER-STEGO PAIR CONSTRAINT, THEN REFINED WITHOUT THE PAIR CONSTRAINT.

Model	UERD			J-UNIWARD			J-MiPOD			Mixture
QF	75	90	95	75	90	95	75	90	95	
MN-xL (Mish)	0.9577	0.9675	0.9570	0.8873	0.8895	0.8919	0.9827	0.9794	0.8621	0.9322
B4 (Mish)	0.9583	0.9664	0.9538	0.8885	0.8878	0.8967	0.9828	0.9807	0.8696	0.9331
B5 (Mish)	**0.9606**	0.9691	**0.9597**	0.8911	0.8945	**0.9025**	0.9851	0.9794	0.8693	0.9360
B6 (Mish)	0.9591	0.9665	0.9567	0.8935	0.8979	0.9022	0.9842	0.9801	**0.8724**	0.9361
B7* (Mish)	0.9592	**0.9713**	0.9528	**0.9052**	**0.9112**	0.8937	**0.9876**	**0.9821**	0.8600	**0.9385**

Table II
WAUC FOR FIVE PRE-TRAINED MODELS REFINED FOR STEGANALYSIS WITH MISH ACTIVATION AND ON NON-ROUNDED RGB PIXELS.

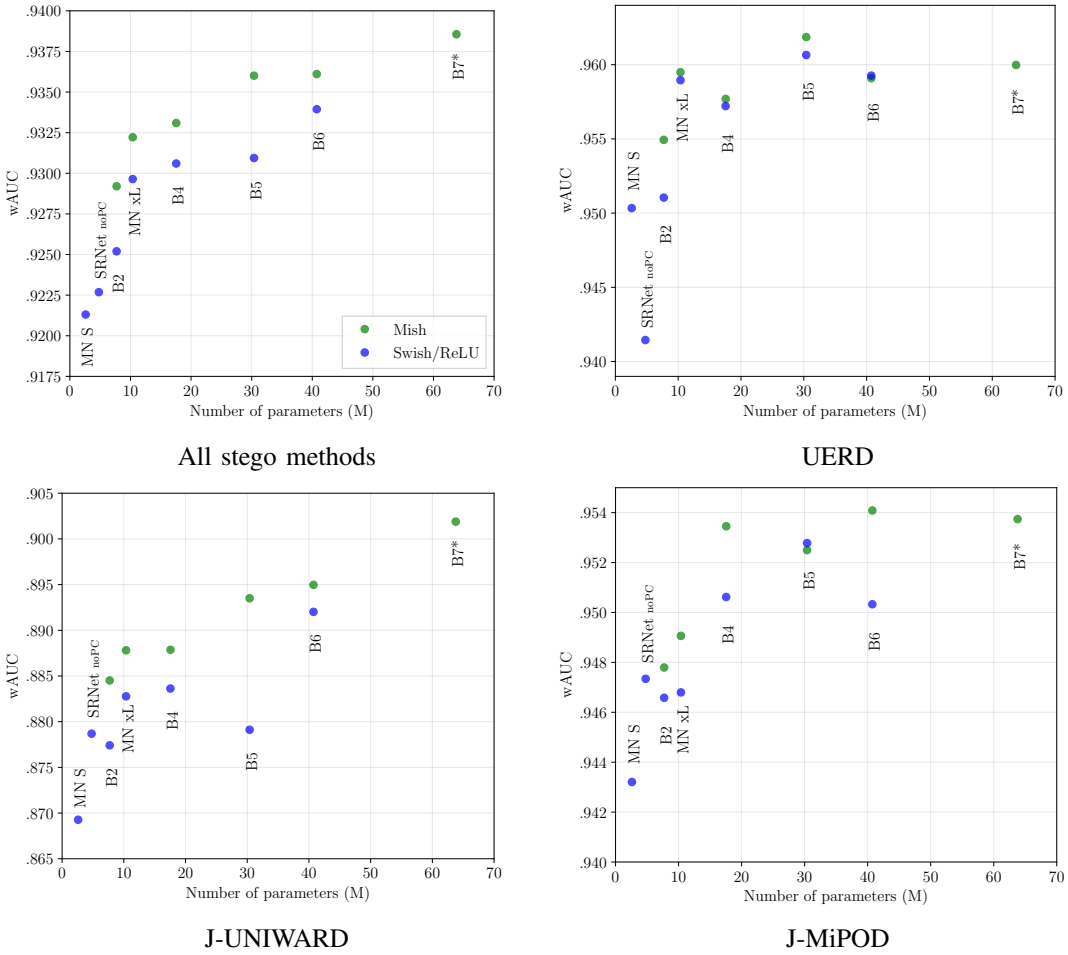

Figure 1. Performance in terms of wAUC versus model size (number of parameters) of SRNet noPC and different ImageNet pre-trained models using Swish (blue) and Mish (green) activation functions.

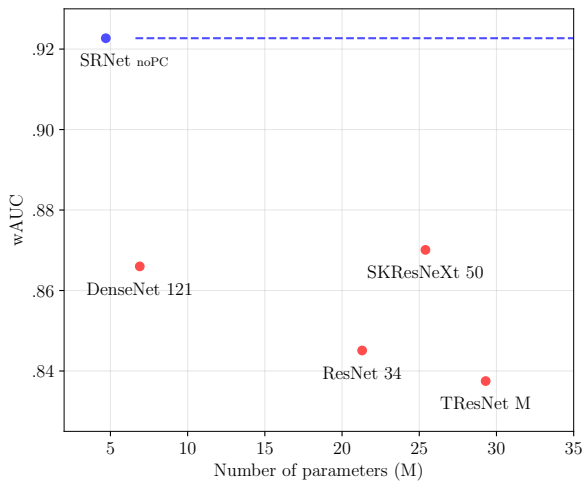

Figure 2. Performance in terms of wAUC versus model size (number of parameters) across different ImageNet pre-trained models building on the ResNet stem and SRNet noPC.

MixNet-S stem	wAUC
3×3 conv, stride 1	**0.9353**
3×3 conv, stride 2	0.9213
3×3 conv, stride 1→3×3 avg pool, stride 2	0.8445
3×3 conv, stride 2→3×3 avg pool, stride 2	0.8046

Table III
WAUC FOR FOUR VARIANTS OF THE MIXNET-S ARCHITECTURE.

consisted of two separate ensembles trained on different splits. Within each ensemble, we trained a 2nd-level stacking model on the detectors' outputs on the validation split (catboost [40] for Ensemble 1, xgboost [41] for Ensemble 2):

- Ensemble 1: QF (target encoded), DCTR, JRM, SRNet, MixNet-S, MixNet-xL (Mish), EfficientNet B2, B4 Mish, B5 (Mish), and B6 (Mish) (1 fold) (test score 0.9401, private score 0.931, public score 0.935)
- Ensemble 2: QF (one hot encoded), EfficientNet B6* (4 folds), B6* (Mish) (2 folds) and B7* (Mish) (2 folds) (test score 0.9424, private score 0.932, public score 0.941)

Due to the small size of the public LB, the best detector in Ensemble 1 (B6 (Mish)) performed surprisingly low on the public LB (public score 0.932), but had one of the best single model performances on the private LB (private score 0.926).

	SRNet	B4 (Mish)	OneHot+SRNet
J-UNIWARD	4.55, 4.66	5.03, 5.14	**4.31, 4.00**
EBS	2.13, 1.20	**1.44, 0.77**	2.47, 1.51
UED	**5.03, 5.49**	5.49, 6.08	5.24, 5.63
nsF5	11.51, 24.60	13.97, 31.47	**3.80, 3.14**

Table IV
DETECTION ERROR, MISSED DETECTION AT 5% FALSE ALARM (P_E, MD_5) OF SRNET, EFFICIENTNET-B4 WITH MISH ACTIVATION, AND ONEHOT+SRNET [39] ON THE ALASKA I DATASET (TILES, QF95, DOUBLE PAYLOAD) WHEN TESTED AGAINST INDIVIDUAL STEGO ALGORITHMS.

We decided not to include it in our final submissions. The final blending was done by rank averaging submissions from Ensemble 1 and 2, which had a private score of 0.932 and public score of 0.944.

VI. CONCLUSIONS

This paper looks into the possibility to build steganalysis detectors from computer vision models pre-trained on ImageNet and refined on examples of cover and stego images. Due to time constraints, our study is limited to the setup of ALASKA II and some selected cases. Besides superior detection accuracy, the pre-trained models offer other significant advantages over models that have to be trained from scratch: the transfer learning is orders of magnitude faster than training a dedicated steganalysis CNN from scratch and is more data efficient. The refining can be done more efficiently and can be done for multiple quality factors at the same time, which drastically reduces the training complexity.

The authors conjecture that the superior detection performance is due to the fact that the pre-trained models have been exposed to a great variety of content and thus are able to better learn noise patterns modulated by content – the stego signal, with less data than specialized CNNs trained from scratch. Preliminary experiments on the ALASKA I dataset show that the accuracy benefit of ImageNet pre-trained models diminishes with more training data.

This paper poses more questions than it answers. Many interesting questions remain, such as the ability of the pre-trained models to generalize to custom JPEG quantization tables, and what the refinement should be for building detectors for spatial domain steganography. The authors are eager to pursue these directions in the near future.

ACKNOWLEDGMENT

The work on this paper was supported by NSF grant No. 1561446 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 DARPA or the U.S. Government.

REFERENCES

[1] Y. Q. Shi, C. Chen, and W. Chen, "A Markov process based approach to effective attacking JPEG steganography," in *Information Hiding, 8th International Workshop* (J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, eds.), vol. 4437 of Lecture Notes in Computer Science, (Alexandria, VA), pp. 249–264, Springer-Verlag, New York, July 10–12, 2006.

[2] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," in *Proceedings of the 11th ACM Multimedia & Security Workshop* (J. Dittmann, S. Craver, and J. Fridrich, eds.), (Princeton, NJ), pp. 75–84, September 7–8, 2009.

[3] 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.

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

[5] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 432–444, April 2012.

[6] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 868–882, June 2011.

[7] J. Kodovský and J. Fridrich, "Steganalysis of JPEG images using rich models," in *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2012* (A. Alattar, N. D. Memon, and E. J. Delp, eds.), vol. 8303, (San Francisco, CA), pp. 0A 1–13, January 23–26, 2012.

[8] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, "Steganalysis of adaptive JPEG steganography using 2D Gabor filters," in *3rd ACM IH&MMSec. Workshop* (P. Comesana, J. Fridrich, and A. Alattar, eds.), (Portland, Oregon), June 17–19, 2015.

[9] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," in *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015* (A. Alattar and N. D. Memon, eds.), vol. 9409, (San Francisco, CA), February 8–12, 2015.

[10] G. Xu, H. Z. Wu, and Y. Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, pp. 708–712, May 2016.

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

[12] M. Yedroudj, F. Comby, and M. Chaumont, "Yedroudj-net: An efficient CNN for spatial steganalysis," in *IEEE ICASSP*, (Alberta, Canada), pp. 2092–2096, April 15–20, 2018.

[13] B. Li, W. Wei, A. Ferreira, and S. Tan, "ReST-Net: Diverse activation modules and parallel subnets-based CNN for spatial image steganalysis," *IEEE Signal Processing Letters*, vol. 25, pp. 650–654, May 2018.

[14] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, "JPEG-phase-aware convolutional neural network for steganalysis of JPEG images," in *The 5th ACM Workshop on Information Hiding and Multimedia Security* (M. Stamm, M. Kirchner, and S. Voloshynovskiy, eds.), (Philadelphia, PA), June 20–22, 2017.

[15] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 1181–1193, May 2019.

[16] J. Zeng, S. Tan, G. Liu, B. Li, and J. Huang, "WISERNet: Wider separate-then-reunion network for steganalysis of color images," *CoRR*, vol. abs/1803.04805, 2018.

[17] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1138–1150, 2020.

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

[19] J. Yang, Y.-Q. Shi, E. Wong, and X. Kang, "JPEG steganalysis based on densenet," *CoRR*, vol. abs/1711.09335, 2017.

[20] 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*, vol. 13, pp. 1200–1214, May 2018.

[21] M. Chen, M. Boroumand, and J. Fridrich, "Reference channels for steganalysis of images with convolutional neural networks," in *The 7th ACM Workshop on Information Hiding and Multimedia Security* (R. Cogranne and L. Verdoliva, eds.), (Paris, France), ACM Press, July 3–5, 2019.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE conference on computer vision and pattern recognition*, pp. 248–255, June 20–25, 2009.

[23] S. Ozcan and A. F. Mustacoglu, "Transfer learning effects on image steganalysis with pre-trained deep residual neural network model," in *IEEE International Conference on Big Data (Big Data)*, pp. 2280–2287, December 10–13, 2018.

[24] 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)*, pp. 770–778, June 27–30, 2016.

[25] T. Ridnik, H. Lawen, A. Noy, and I. Friedman, "TResNet: High performance GPU-dedicated architecture," *arXiv preprint arXiv:2003.13630*, 2020.

[26] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 510–519, June 16–20, 2019.

[27] 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.

[28] T. Mingxing and V. L. Quoc, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML*, vol. 97, pp. 6105–6114, June 9–15, 2019.

[29] T. Mingxing and V. L. Quoc, "MixConv: Mixed depthwise convolutional kernels," in *British Machine Vision Conference, BMVC*, September 9–12, 2019.

[30] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2661–2671, June 16–20, 2019.

[31] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020.

[32] 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*, vol. 2014:1, 2014.

[33] R. Cogranne, Q. Giboulot, and P. Bas, "Steganography by minimizing statistical detectability: The cases of JPEG and color images," in *The 8th ACM Workshop on Information Hiding and Multimedia Security* (C. Riess and F. Schirrmacher, eds.), (Held virtually), ACM Press, 2020.

[34] L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient JPEG steganography," *IEEE Transactions on Information Forensics and Security*, vol. 9, pp. 814–825, May 2014.

[35] Y. Yousfi and J. Fridrich, "JPEG steganalysis detectors scalable with respect to compression quality," in *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2020* (A. Alattar and N. D. Memon, eds.), (San Francisco, CA), January 26–30, 2020.

[36] D. Misra, "Mish: A self regularized non-monotonic neural activation function," *arXiv preprint arXiv:1908.08681*, 2019.

[37] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[38] R. Cogranne, Q. Giboulot, and P. Bas, "The ALASKA steganalysis challenge: A first step towards steganalysis "Into the wild"," in *The 7th ACM Workshop on Information Hiding and Multimedia Security* (R. Cogranne and L. Verdoliva, eds.), (Paris, France), ACM Press, July 3–5, 2019.

[39] Y. Yousfi and J. Fridrich, "An intriguing struggle of CNNs in JPEG steganalysis and the OneHot solution," *IEEE Signal Processing Letters*, vol. 27, pp. 830–834, 2020.

[40] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Advances in neural information processing systems*, pp. 6638–6648, December 3–8, 2018.

[41] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, August 13–17, 2016.