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 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 into 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.

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 sub-model (Figure 2) separately reveals that most of the detection performance is due to the sub-model ’Ax_T5’ corresponding to integral co-occurrences 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.
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 $M$ is first clipped to a threshold $T$ and then transformed to a binary volume of size $(T+1) \times H \times W$:

$$ Z^{H \times W} \rightarrow \{0, 1\}^{(T+1) \times H \times W} $$

$$ M \mapsto \begin{cases} \lfloor |M| = t \rfloor, & t \in \{0, ..., T - 1\} \\ \lfloor |M| \geq t \rfloor, & t = T \end{cases} \quad (1) $$

where $[\cdot]$ is the (element-wise) Iverson bracket $[\cdot]$. 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 $K \in \mathbb{R}^{(T+1) \times 3}$, followed by global average pooling

$$ K = \begin{cases} K_t = 0 & t = x \\ K_t = 0 & t = y \\ K_t = 0_{3 \times 3} & \text{else.} \end{cases} \quad (2) $$

Inter-block statistics can be designed similarly by using dilated convolutions with rate 8 introduced in wavelet decompositions algorithms [15], also referred to as “a 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 $(D_1, D_2) = (64, 0)$ 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
Training FLD ensemble. This strategy is denoted JRM+SRNet. SRNet’s last layer and JRM features as a feature set and Hot+SRNet compares to simply concatenating (a trained) are merged and how the resulting architecture denoted One-Hot+SRNet detects. We use SRNet to show how these two networks with conventional CNN architectures produces more universal detectors. 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 other steganographic schemes [17].

We show that cartesian calibration helps nsF5, Jsteg, YASS, and some large degradations of JRM+SRNet (e.g., J-UNIWARD, UNIWARD and UED-JC [14] in JPEG round while avoiding has comparable detection performance to SRNet for J-RM+SRNet. The next two blocks show that OneHot+SRNet in JPEG trunc, the improvement is consistently better than substantially improves SRNet. For nsF5 in JPEG round, the invariance losses as follows:

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

where \(\lambda_B\) 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 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 \(p_B\) (the output of the \(FC_B+\text{softmax}\)). \(\mathcal{F}_{SRNet}\) and \(\mathcal{F}_{OneHot}\) are then concatenated and fed to the final FC layer, which outputs \(p_{FC}\), the final classification probability.

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

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

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.

<table>
<thead>
<tr>
<th>QF</th>
<th>100</th>
<th>95</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round, nsF5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>OneHot CNN</td>
<td>3.49</td>
<td>20.65</td>
<td>7.90</td>
</tr>
<tr>
<td>Trunc, J-UNI</td>
<td>0.4</td>
<td>0.3</td>
<td>0.4</td>
</tr>
<tr>
<td>OneHot CNN</td>
<td>10.81</td>
<td>19.60</td>
<td>15.38</td>
</tr>
</tbody>
</table>

Table III: Detection error \(P_E\) of ccJRM and the calibrated OneHot CNN for JPEG round, nsF5 for various quality factors and payloads.

<table>
<thead>
<tr>
<th>QF</th>
<th>100</th>
<th>95</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round, nsF5</td>
<td>2.11</td>
<td>18.03</td>
<td>7.15</td>
</tr>
<tr>
<td>ccJRM</td>
<td>1.39</td>
<td>15.52</td>
<td>6.66</td>
</tr>
<tr>
<td>ccOneHot CNN</td>
<td>4.17</td>
<td>20.65</td>
<td>7.90</td>
</tr>
</tbody>
</table>

Figure 4. OneHot CNN architecture. \(D_i\) corresponds to the depth of representations at different layers of the network.
v1 and QF 95, OneHot+SRNet tile detector performs cases when the OneHot branch is not effective. For ALASKA the reported struggles while not decreasing the performance in (SRNet) for an end-to-end trainable detector. This overcomes for adding the OneHot CNN to existing CNNs for steganalysis flexible learnable manner.

computing higher-order statistics of DCT coefficients in a in at least two particular scenarios reported in this paper.

In steganalysis. In this work, a new CNN architecture is underperforming, for example, in solving the seemingly trivial results.

use of size each channel using the same threshold, producing a volume one-hot encoding" in the OneHot branch is done separately for layer by using

detectors. Color channels for SRNet are merged in the first quality factor 95. As prescribed by the challenge winners [28], v1

Round, UED-JC

OneHot+SRNet

OneHot+SRNet

Round, UED-JC

OneHot+SRNet

OneHot+SRNet

Round, UED-JC

OneHot+SRNet

Table IV

Table V

Table VI

<table>
<thead>
<tr>
<th>QF</th>
<th>100</th>
<th>95</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round, nsF5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>JRM+SRNet</td>
<td>13.88</td>
<td>30.76</td>
<td>12.63</td>
</tr>
<tr>
<td>OneHot+SRNet</td>
<td>1.97</td>
<td>18.11</td>
<td>3.50</td>
</tr>
<tr>
<td>Trunc, J-UNI</td>
<td>1.99</td>
<td>18.55</td>
<td>3.32</td>
</tr>
<tr>
<td>Round, J-UNI</td>
<td>0.4</td>
<td>0.3</td>
<td>0.4</td>
</tr>
<tr>
<td>JRM+SRNet</td>
<td>16.37</td>
<td>21.26</td>
<td>30.26</td>
</tr>
<tr>
<td>OneHot+SRNet</td>
<td>7.62</td>
<td>17.49</td>
<td>14.32</td>
</tr>
<tr>
<td>JRM+SRNet</td>
<td>7.29</td>
<td>13.64</td>
<td>14.18</td>
</tr>
<tr>
<td>OneHot+SRNet</td>
<td>11.94</td>
<td>16.99</td>
<td>17.52</td>
</tr>
<tr>
<td>Round, UED-JC</td>
<td>0.3</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>JRM+SRNet</td>
<td>6.96</td>
<td>10.16</td>
<td>10.90</td>
</tr>
<tr>
<td>OneHot+SRNet</td>
<td>7.69</td>
<td>10.53</td>
<td>10.99</td>
</tr>
<tr>
<td>OneHot+SRNet</td>
<td>7.26</td>
<td>10.58</td>
<td>11.35</td>
</tr>
</tbody>
</table>

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 × 3 × 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) × H × 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 Mat-lab’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 $M \in \mathbb{Z}^{H \times W}$ thanks to the symmetries of DCT bases. We denote $f_b(M)$ any operation $f$ performed in a block-wise fashion, with a block size of 8, e.g., $T_b$ is the $8 \times 8$ block-wise matrix transpose. For simplicity, we introduce $J = [(−1)^i]_{0 \leq i < H}$ and $I = [(−1)^j]_{0 \leq j < W}$. 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 $\pi/2$

\[
\text{Lossless flip}_Y (M) = J \cdot \text{flip}_Y \circ \text{flip}_V (M) \tag{4}
\]

\[
\text{Lossless rot}_{\pi/2} (M) = I \cdot T_b \circ \text{rot}_{\pi/2} (M) \tag{5}
\]

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


