Towards Order of Processing Operations Detection in JPEG-compressed Images with Convolutional Neural Networks

Belhassen Bayar and Matthew C. Stamm; Drexel University, Philadelphia, PA, bb632@drexel.edu, mstamm@coe.drexel.edu

Abstract

Determining which processing operations were used to edit an image and the order in which they were applied is an important task in image forensics. Existing approaches to detecting single manipulations have proven effective, however, their performance may significantly deteriorate if the processing occurs in a chain of editing operations. Thus, it is very challenging to detect the processing used in an ordered chain of operations using traditional forensic approaches. First attempts to perform order of operations detection were exclusively limited to a certain number of editing operations where feature extraction and order detection are disjoint. In this paper, we propose a new data-driven approach to jointly extract editing detection features, detect multiple editing operations, and determine the order in which they were applied. We design a constrained CNN-based classifier that is able to jointly extract low-level conditional fingerprint features related to a sequence of operations as well as identify an operation's order. Through a set of experiments, we evaluated the performance of our CNN-based approach with different types of residual features commonly used in forensics. Experimental results show that our method outperforms the existing approaches.

Introduction

Identifying the processing history of an image has been investigated over the last decade given its importance in wide variety of settings [25]. Digital images are used as evidence in legal proceedings and criminal investigations. Therefore, it is important to determine the types of editing operation that an image has undergone and the order in which they have been applied. This can provide a forensic analyst a complete information about a forged image, such as the multiple types of processing operations frequently used by a forger, and may enlighten directions to determine the party who created the forgery.

Image editing operations typically leave behind unique artifacts, known as fingerprints, that can be used to determine its type. Early approaches proceeded by extracting representative features of these fingerprints then developed associated algorithms to determine the type of a particular image tampering. This approach has proven effective in detecting many types of image tampering such as median filtering [19, 17], contrast enhancement [24], resizing and resampling [21, 18], etc.

While exiting forensic approaches can determine if and how an image has undergone a single processing operation, their performance typically deteriorates if the processing occurs as a sequence of image editing operations [23, 8]. This is because fingerprints produced by latter processing operations can destroy processing fingerprints left by a former processing operation [23, 8]. Furthermore, detecting the order of operations can give a forensic investigator a complete picture of the global processing operations in order to understand to which extend a subject image has been altered from its raw version captured by a camera.

Little work has been done to determine the order of a processing operation within a sequence of image editing. Early approaches proposed the notion of conditional fingerprints to determine the order of an operation occurring in an ordered chain of image editing. More specifically, Stamm *et. al* [23] proposed to differentiate between the ordered chains using a sequence of intermediate grouped hypothesis tests. Then, each intermediate stage searches for the presence of a specific fingerprint or conditional fingerprint. However, human discovery of conditional fingerprints and complex hierarchical hypothesis test must be designed for each chain, which is difficult and time consuming. This problem has been also studied from the information-theoretical perspective in [10].

An alternative steganalytic approach has been proposed by Boroumand and Fridrich [8] where a rich model features based approach with quantization step 1 (SRMQ1) [12] was used to determine an operations order. This method operates by building local models of pixel dependencies by analyzing the joint distribution of pixel value prediction-errors. While this approach has proven effective, it still requires a forgery analyst to separately extract features then perform order of operations detection. Thus, several questions remain open: How should low-level predictive feature extractors be designed? Can order of operation detection features be learned directly from data? Can we devise a generic approach that can jointly extract features and perform order of operations detection?

Furthermore, in a realistic scenario images are typically JPEG compressed to be saved after being processed. JPEG compression makes order of processing operation detection problem harder because a substantial component of the processing artifacts for forensic purposes is significantly suppressed or distorted. Our ultimate goal is to devise a robust forensic approach to JPEG compression that can jointly (1) learn pixel-value relationship traces left by editing processing chains and (2) determine the order in which these editing operations have been applied.

In this paper, we propose a more practical data-driven approach to perform order of operations detection that is able to distinguish between conditional fingerprints left by different ordered chains of editing operations. To accomplish this, we employed a constrained convolutional neural network (CNN) based approach which can adaptively learn low-level prediction error features directly from data. In fact, CNNs tend to learn content-dependent features from images.Therefore, researchers in forensics employed several prediction error feature extractors associated with the CNNs. In our experiments, we evaluated our CNN-based approach with different types of prediction error feature extractors commonly used in forensics, i.e., the adaptive constrained convolutional layer used in forensics [1], the median filter residual (MFR) features employed in median filtering detection [9], and the high-pass filter (HPF) first adopted in steganalysis [20].

In what follows, we provide an overview of our proposed data-driven approach including details on the employed contsrained CNN architecture. Next, we evaluated our approach to perform order of processing operations detection with three different types of image editing operations. Additionally, we studied the impact of the training data size on CNN's performance. Results of these experiments showed that our approach outperforms the state-of-the-art method with JPEG re-compressed images and can achieve 96.38% with three processing operations using a large scale training dataset.

Proposed method

The goal of this paper is to devise a more practical data-driven approach to performing order of processing operations detection in JEPG re-compressed images. As noted above, early approaches rely on theoretical analysis and parametric models of image's data [8, 23] which may not be accurate enough particularly in challenging scenarios. More specifically, in realistic scenarios images are in general JPEG compressed such as in social media and photo-sharing websites. This makes forensic traces very difficult to detect, hence, the performance of traditional forensic approaches significantly deteriorates [8].

Typically, a forger uses a sequence of several editing operations to create a forgery, also called a chain of processing operations. It is very challenging to detect chains of editing operations using traditional approaches. This is because operations that occur later in the processing sequence can potentially destroy or alter fingerprints left by operations that occur earlier in the sequence. Moreover, as mentioned above traditional approaches rely on theoretical model of image's data [8, 23] which may not be accurate enough to detect manipulation fingerprints left by a chain of processing operation.

Chains of processing operations are associated with an order in which a forger employs to perform a sequence of manipulations within an image. These chains leave behind manipulation traces, known as conditional fingerprints [23] , which are dependent on the employed order of operations in the chain. It is very challenging to detect traces induced by every single editing operation and determine its corresponding order in a processing chain. Therefore, forensic investigators must first measure and analyze traces left by the cumulative effect induced by an ordered processing operations chain [8].

Instead of relying on theoretical analysis of parametric models, we propose a data-driven approach to directly learn from data the cumulative effect induced by an ordered chain of processing operations. To accomplish this, we cast the order of processing operations detection as a classification problem where every processing chain corresponds to a class. To better understand this, Fig. 1 illustrates two chains of processing operations that consist of the same set of N editing operations. If we invert the order of operations i and j in chain 1, this will correspond to a different class of ordered chain 2 that leaves behind different and unique cumulative effect in an image. Our approach works for general scenarios such that the total number of operations N that a chain consists of is arbitrary and can be as low as 1. Furthermore, our method requires a forensic investigator to assign a label to each different ordered processing chain as a separate class in the training data.

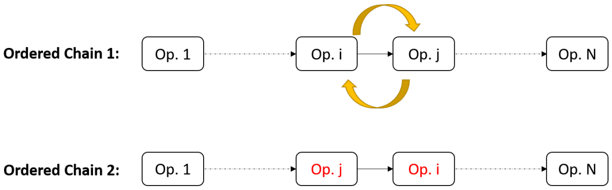

Figure 1: Two chains of the same N processing operations using different order; Chains 1 and 2 leave behind different traces.

To formulate this problem, let us consider the detection problem where a forgery analyst would like to build a forensic detector algorithm $g(\cdot)$ that identifies the type of editing operation which an image X has undergone. Typically, the forensic detector g is designed as the composition of two functions $f(\cdot)$ and $d(\cdot)$ such that

$$g(\mathrm{X}) = d \circ f(\mathrm{X}) = d(f(\mathrm{X})), \tag{1}$$

where $f(\cdot)$ is a feature extractor and $d(\cdot)$ is the discriminative classifier that determines the type of processing operation applied to an image X. In general, traditional forensic approaches employ a detection procedure where feature extraction and detection are separate, i.e., functions f and d are completely disjoint.

In this paper, we employ the constrained CNN [3] as our forensic detection system g in order to jointly learn forensic features directly from data and identify the order of processing operations in JPEG re-compressed images. Constrained CNNs have proven effective at performing several multimedia forensic tasks [3, 2, 5, 4]. These type of deep neural networks enforce the CNN to learn prediction error filters at the first convolutional layer while training. Prediction error filters in CNN suppress an image's content and can adaptively learn from data low-level pixel relationships induced by different types of image editing operations.

In our method, the feature extraction function f corresponds to CNN's layers, before the classification layer, which learn deep hierarchical forensic features throughout the network. We use the last fully-connected layer of the CNN followed by a softmax associated with an arg max operator as the forensic detector. In order to devise a more powerful forensic detector d, we employ the deep features approach [6] commonly used in computer vision [11]. To accomplish this, we extract the activated deep features $f(\mathrm{X})$ from our constrained CNN. Next, we compute a confidence score s_k for each k^{th} class by training an extremely randomized trees (ERT) classifier [13] using the deep forensic features $f(\mathrm{X})$ after activation. Next, we predict the processing operations order using our new ERT-based detection system

$$g(\mathrm{X}) = \arg\max_{m_k \in \mathscr{C}} s_k\big(f(\mathrm{X})\big) = \hat{m}, \tag{2}$$

where the detector d consists of the ERT-based confidence scores s_k's associated with the arg max operator, $\hat{m}$ is the predicted class (i.e., ordered chain), and $\mathscr{C}$ is the set of all possible types of image editing including the unaltered images class.

Unlike traditional approaches which rely on hand-designed features, our constrained CNN can learn directly from data conditional forensic fingerprints [23] related to the order of processing

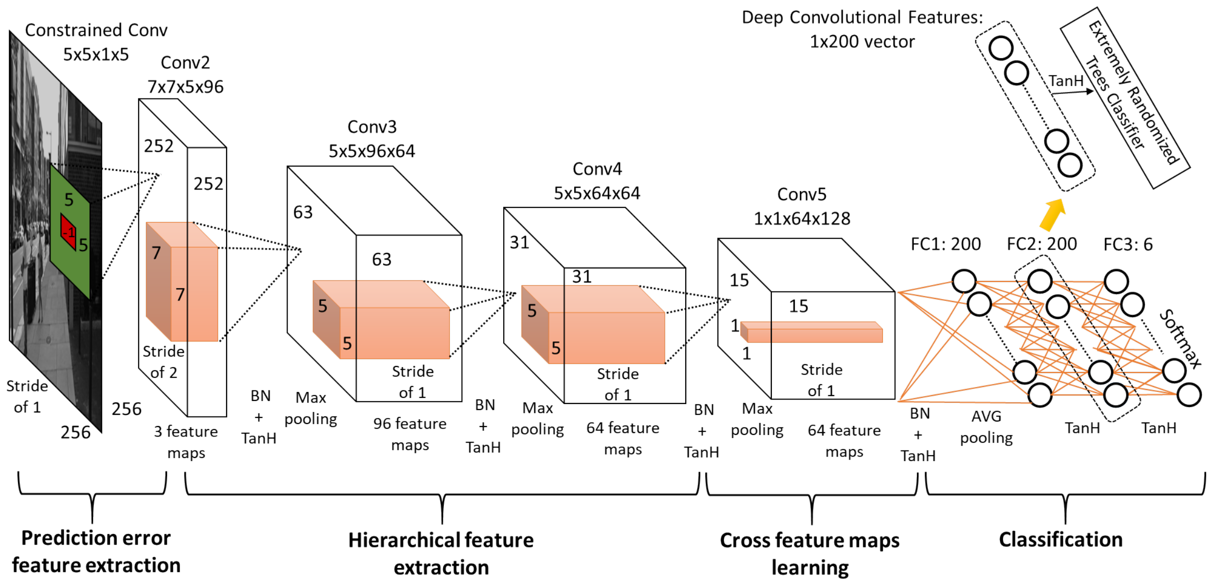

Figure 2: CNN proposed architecture; BN:Batch-Normalization Layer; TanH: Hyperbolic Tangent Layer

operations. Furthermore, several approaches to order of processing operations detection are designed using targeted forensic detector [23] which is difficult and time consuming. By contrast, our method is generic and scalable to all different types and number of applied image editing operations.

Detector architecture

In this section, we give a brief overview about our proposed CNN architecture. Fig. 2 depicts the four different designed conceptual blocks that we used in our CNN as well as the dimension of each layer and its corresponding output. The input layer corresponds to a grayscale 256×256 pixel sized image. In what follows, we describe in detail each used conceptual block in our CNN architecture.

Pixel-value dependency feature extraction: As mentioned above, the existing CNNs tend to learn content-dependent features from images which is very well-suited for object recognition tasks. If CNNs of this form are used to perform order of operations detection, this may lead to a classifier that identifies objects associated with the training data as opposed to learn traces left by an ordered sequence of operations. To overcome this problem, we used a constrained convolutional layer [1] in the first layer of our CNN architecture. This type of layer has proven effective at extracting generic image manipulation features [1] which take the form of low-level pixel-value dependency features. In total, we used five different 5×5 constrained convolutional filters which will produce prediction residual feature maps of size 252×252×5.

Hierarchical feature extraction: In order to learn higher-level classification features, one can notice from Fig. 2 that we used a conceptual block which consists of three different regular convolutional layers. Each of these layers learns new representation of the data and is followed by a batch normalization (BN) layer [15], a hyperbolic tangent (TanH) activation function, and a max-pooling layer. Furthermore, we can notice that the prediction residual feature maps produced by the previous conceptual block are directly passed to a regular convolutional layer in the hierarchical feature extraction block. This is because these type of features are vulnerable to be destroyed by nonlinear operations such as pooling and TanH activation function [3].

Cross feature maps learning: Convolutional layers in CNN are able to learn new associations between features within the same feature map. In our CNN architecture the output of the hierarchical feature extraction conceptual block is fed to a 1×1 convolutional layer that consists of 128 filters. This type of layer will learn new associations of features located at the same spatial location but in a different feature map. The 1 × 1 convolutional layer has shown to improve CNN's performance when applied to a steganalysis task [26]. Finally, the output of the 1×1 convolutional layer is followed by an average pooling layer. In our experiments, the choice of average pooling has empirically demonstrated to outperform other types of pooling layers when used after the 1×1 convolutional layer.

Classification: The last conceptual block in our CNN architecture consists of three fully-connected layers. Similarly to the convolutional layers, the first two fully-connected layers are followed by a TanH activation function and each contains 200 neurons. These two layers are used to learn new associations of the highest-level convolutional features learned by the former block. Finally, the output layer, known as classification layer, is followed by a softmax. This type of activation function maps features learned by a CNN to a set of probability values where neurons in this layer sum to one. The total number of neurons in the classification layer is equal to the total number of image manipulations we a consider in a detection task including the unaltered images class. Each input image will correspond to the class associated with the highest activated neuron. Finally as described in the previous section, we improve the performance of our CNN by using the deep features approach [11]. To do this, we extract the deep features from the second fully-connected of our CNN by doing a feedforward pass of our data. Next, we train an ERT classifier to detect the order of processing operations in altered images as described in Eq. (2).

Table 1: Used editing operations to create our experimental databases.

Editing operation	Parameter
Unaltered (UA)	–
Median Filtering (MF)	$K_{size} = 5\times5$
Gaussian Blurring (GB) with $\sigma = 1.1$	$K_{size} = 5\times5$
Resampling (RS) with bilinear interpolation	Scaling $= 1.5$

General experimental setup

We assessed the performance of our proposed constrained CNN-based approach to perform order of operations detection through a set of experiments. We conducted three different experiments: (1) first we evaluated CNNs performance with different residual feature extractors, (2) we then compared our constrained CNN-based method to the spatial rich model features approach [8] associated with the ensemble classifier [12], (3) finally we assessed the impact of the training set size on CNN's performance.

To accomplish these experiments, we collected 15,125 images from the publicly available Dresden Image Database [14] to create several experimental databases. In order to generate grayscale images, we retained the green layer of the 16 central 256×256 patches. Images were manipulated using three processing operations listed in Table 1. For ease of analysis, each chain of operations can consist up to two processing operations.

We adopt the notation X-Y to denote a sequence where the patch was first edited using manipulation X, then subsequently edited using manipulation Y (i.e. MF-RS corresponds to first applying median filtering, then applying resizing). Additionally, we considered two different scenarios where images have been manipulated with and without *redundant* processing operations (i.e. MF-MF corresponds to first applying median filtering, then applying the same median filtering operation).

In all our experiments, CNNs were trained for 36 epochs. We set the batch size equal to 64 and the parameters of the stochastic gradient descent as follows: $momentum = 0.95$, $decay = 0.0005$, and a learning rate $\varepsilon = 10^{-3}$ that decreases every three epochs by a factor $\gamma = 0.5$. Additionally, while training CNNs, their testing accuracies on a separate testing dataset were recorded every 1,000 iterations to produce tables and figures in this section. Note that training and testing are disjoint. We implemented all of our CNNs using the Caffe deep learning framework [16]. We ran our experiments using an Nvidia GeForce GTX 1080 GPU with 8GB RAM. The datasets used in this work were all converted to the lmdb format.

CNN-based residual features analysis

In our first set of experiments, we evaluated the performance of our CNN to perform order of processing operations detection when it is associated with different prediction residual feature extractors commonly used in forensics, i.e., the constrained convolutional layer [1], the high-pass filter (HPF) used in [20, 3], and the median filter residual (MFR) used in [9, 2]. In this part, we excluded the redundant operations scenario. Additionally, to evaluate the impact of the JPEG re-compression on the CNN's performance we considered two scenarios when images were JPEG re-compressed as well as uncompressed.

To conduct these experiments, we created an experimental grayscale image database that consisted of 296,000 training patches and 52,000 testing patches of size 256×256 in the same manner described above using 1,882 images. Training and testing patches were created from two separate sets of images and were edited using three types of image manipulations listed in Table 1 along with their corresponding all possible pairs of editing. This resulted in 10 different processing classes including the unaltered patches. Next, each grayscale patch was JPEG re-compressed using quality factor of 90.

Table 2: Residual feature analysis using our proposed CNN with and without JPEG re-compression after processing.

Method	JPEG (QF=90)	No compression
Proposed	92.90%	96.38%
HPF-based CNN	80.63%	92.99%
MFR-based CNN	72.54%	87.17%

Table 3: Order of operations detection rate in JPEG-compressed (QF=90) images. Redundant operations were excluded.

Operations	SRMQ1	CNN	CNN (ERT)
w/out redundant op.	93.76%	92.90%	**94.19%**
w/ redundant op.	85.82%	85.66%	**87.12%**

Table 2 depicts the identification rate of CNN when associated with the three different choices of prediction error feature extractors. From Table 2, one can observe that our proposed CNN associated with a constrained convolutional layer outperforms the other choices of feature extractors (i.e., HPF and MFR based CNNs) and can achieve 96.38% identification rate in uncompressed images. Additionally, we can notice that when the processing operations are followed by a JPEG compression our constrained CNN can achieve 92.90% identification rate and outperforms the other choices of CNN by at least 12.27%. The constrained CNN has proven robust to the JPEG scenario since its performance decreases by only 3.48% identification rate. By contrast, CNN's performance significantly deteriorates with other choices of feature extractor and it can achieve 80.63% identification with HPF and 72.54% identification rate with MFR feature extractor.

These results demonstrate the advantage of using the constrained convolutional layer. In fact, the HPF and MFR based approaches achieve a lower detection rate since they are a suboptimal solution of the trained network with a constrained convolutional layer. Thus, the constrained convolutional layer can capture image manipulation features that may not be captured using a fixed feature extractor and can lead to significantly better performance. This has been demonstrated particularly when editing operations are followed by JPEG compression.

Comparison with SRM-based approach

We compared our proposed CNN method to the spatial rich model (SRMQ1) features [8] based approach using the ensemble classifier in [12] in two scenarios where images were edited using redundant and non redundant processing operations. To accomplish this, we first used the same experimental database that we created in the previous set of experiments for the non redundant operations scenario using the re-compressed image patches. Next, we created a second experimental database for the redundant processing operations scenario in the same manner described above.

Table 4: Confusion matrix for identifying the order of processing operations using our ERT-based constrained CNN in JPEG re-compressed images (QF=90) without redundant operations.

Predicted Class

True Class

	UA	MF	GB	RS	MF-GB	GB-MF	MF-RS	RS-MF	GB-RS	RS-GB
UA	**99.33%**	0.06%	0.1%	0.05%	0.00%	0.00%	0.02%	0.00%	0.00%	0.00%
MF	0.15%	**91.77%**	0.02%	0.02%	0.52%	2.12%	0.29%	5.10%	0.00%	0.02%
GB	0.00%	0.21%	**95.00%**	0.87%	0.42%	0.04%	0.04%	0.00%	0.06%	3.37%
RS	0.17%	0.00%	0.65%	**98.94%**	0.02%	0.00%	0.08%	0.08%	0.02%	0.04%
MF-GB	0.00%	0.31%	0.19%	0.00%	**95.87%**	2.48%	0.02%	0.29%	0.42%	0.42%
GB-MF	0.00%	1.44%	0.00%	0.02%	3.38%	**86.02%**	0.13%	8.87%	0.06%	0.08%
MF-RS	0.02%	0.04%	0.00%	0.01%	0.00%	0.08%	**99.17%**	0.38%	0.21%	0.00%
RS-MF	0.00%	3.54%	0.02%	0.00%	0.17%	9.38%	0.65%	**86.00%**	0.17%	0.06%
GB-RS	0.00%	0.02%	0.00%	0.00%	0.40%	0.02%	0.96%	0.06%	**96.69%**	1.85%
RS-GB	0.00%	0.10%	2.08%	0.06%	0.63%	0.13%	0.04%	0.23%	3.56%	**93.17%**

Table 5: Confusion matrix for identifying the order of processing operations using SRMQ1 based approach [8] in JPEG re-compressed images (QF=90) without redundant operations.

Predicted Class

True Class

	UA	MF	GB	RS	MF-GB	GB-MF	MF-RS	RS-MF	GB-RS	RS-GB
UA	**99.77%**	0.02%	0.02%	0.17%	0.00%	0.00%	0.00%	0.00%	0.02%	0.00%
MF	0.10%	**93.27%**	0.00%	0.04%	0.69%	1.71%	0.25%	3.79%	0.08%	0.08%
GB	0.00%	0.00%	**92.08%**	1.58%	0.12%	0.00%	0.02%	0.00%	0.15%	6.06%
RS	0.44%	0.02%	0.98%	**97.98%**	0.04%	0.00%	0.00%	0.00%	0.12%	0.42%
MF-GB	0.00%	0.31%	0.27%	0.06%	**96.48%**	1.21%	0.21%	0.08%	0.67%	0.71%
GB-MF	0.00%	1.73%	0.04%	0.02%	1.35%	**87.04%**	0.48%	9.29%	0.06%	0.00%
MF-RS	0.02%	0.15%	0.00%	0.12%	0.08%	0.12%	**99.12%**	0.31%	0.06%	0.04%
RS-MF	0.00%	3.71%	0.00%	0.06%	0.23%	11.21%	1.02%	**83.65%**	0.08%	0.04%
GB-RS	0.00%	0.33%	0.08%	0.02%	0.56%	0.00%	0.35%	0.00%	**96.62%**	2.06%
RS-GB	0.00%	0.27%	3.40%	0.13%	0.90%	0.00%	0.00%	0.00%	3.67%	**91.62%**

Table 6: Confusion matrix for identifying the order of processing operations using our ERT-based constrained CNN in JPEG re-compressed images (QF=90) with redundant operations.

Predicted Class

True Class

	UA	MF	GB	RS	MF-MF	MF-GB	GB-MF	MF-RS	RS-MF	GB-GB	GB-RS	RS-GB	RS-RS
UA	**99.78%**	0.00%	0.00%	0.19%	0.00%	0.00%	0.03%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MF	0.03%	**44.86%**	0.00%	0.00%	46.78%	0.66%	0.03%	0.19%	1.62%	0.28%	5.47%	0.00%	0.06%
GB	0.00%	0.00%	**89.47%**	0.28%	0.00%	5.62%	0.00%	0.12%	0.03%	0.00%	0.00%	0.00%	4.47%
RS	0.03%	0.06%	0.47%	**98.34%**	0.00%	0.00%	0.97%	0.00%	0.00%	0.06%	0.00%	0.00%	0.06%
MF-MF	0.03%	44.89%	0.00%	0.00%	**46.78%**	0.66%	0.03%	0.19%	1.62%	0.28%	5.47%	0.00%	0.06%
MF-GB	0.00%	0.00%	0.12%	0.00%	0.00%	**92.34%**	0.00%	0.84%	0.03%	0.00%	0.00%	1.03%	5.62%
GB-MF	0.00%	0.00%	0.06%	0.06%	0.00%	0.00%	**96.69%**	0.03%	0.03%	0.03%	0.03%	1.75%	1.31%
MF-RS	0.00%	0.00%	0.06%	0.00%	0.06%	7.00%	0.00%	**89.72%**	2.16%	0.00%	0.25%	0.38%	0.38%
RS-MF	0.00%	0.38%	0.03%	0.00%	0.69%	0.34%	0.00%	3.78%	**83.50%**	0.16%	11.03%	0.00%	0.09%
GB-GB	0.00%	0.09%	0.00%	0.06%	0.03%	0.00%	0.16%	0.00%	0.03%	**99.31%**	0.25%	0.06%	0.00%
GB-RS	0.00%	1.00%	0.00%	0.00%	1.19%	0.03%	0.19%	0.12%	8.09%	0.88%	**88.28%**	0.03%	0.19%
RS-GB	0.00%	0.00%	0.00%	0.00%	0.00%	0.44%	0.19%	0.06%	0.00%	1.16%	0.03%	**96.88%**	1.25%
RS-RS	0.00%	0.00%	0.16%	0.00%	0.00%	9.62%	0.22%	0.12%	0.00%	0.00%	0.16%	2.00%	**87.72%**

Table 7: Confusion matrix for identifying the order of processing operations using SRMQ1 based approach [8] in JPEG re-compressed images (QF=90) with redundant operations.

		Predicted Class												
		UA	MF	GB	RS	MF-MF	MF-GB	GB-MF	MF-RS	RS-MF	GB-GB	GB-RS	RS-GB	RS-RS
True Class	UA	**99.93%**	0.00%	0.00%	0.06%	0.00%	0.00%	0.03%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	MF	0.00%	**26.46%**	0.00%	0.00%	68.21%	0.12%	0.09%	0.31%	1.40%	0.25%	3.00%	0.06%	0.03%
	GB	0.00%	0.00%	**91.75%**	1.40%	0.00%	3.37%	0.06%	0.03%	0.00%	0.03%	0.00%	0.06%	3.28%
	RS	0.25%	0.00%	0.93%	**98.00%**	0.00%	0.00%	0.65%	0.00%	0.00%	0.00%	0.00%	0.00%	0.15%
	MF-MF	0.00%	26.12%	0.00%	0.00%	**68.59%**	0.12%	0.09%	0.31%	1.40%	0.25%	3.00%	0.06%	0.03%
	MF-GB	0.00%	0.06%	2.31%	0.00%	0.00%	**87.96%**	0.00%	1.06%	0.03%	0.03%	0.00%	1.15%	7.37%
	GB-MF	0.00%	0.03%	0.12%	0.40%	0.00%	0.00%	**98.34%**	0.03%	0.00%	0.03%	0.00%	0.56%	0.46%
	MF-RS	0.00%	0.28%	0.15%	0.06%	0.03%	0.81%	0.00%	**96.15%**	1.15%	0.25%	0.06%	0.53%	0.50%
	RS-MF	0.00%	1.12%	0.03%	0.00%	0.50%	0.00%	0.00%	1.81%	**86.65%**	0.46%	9.40%	0.00%	0.00%
	GB-GB	0.00%	0.15%	0.00%	0.06%	0.03%	0.00%	0.31%	0.00%	0.00%	**99.00%**	0.34%	0.06%	0.03%
	GB-RS	0.00%	1.90%	0.00%	0.03%	1.25%	0.00%	0.09%	0.21%	10.90%	0.87%	**84.68%**	0.00%	0.03%
	RS-GB	0.00%	0.34%	0.00%	0.00%	0.09%	1.00%	0.75%	0.37%	0.03%	0.43%	0.00%	**95.65%**	1.31%
	RS-RS	0.00%	0.03%	2.87%	0.03%	0.00%	10.50%	0.65%	0.78%	0.00%	0.00%	0.00%	2.62 %	**82.50%**

To do this, we built a grayscale image database that consisted of 769,600 training patches and 41,600 testing patches pixel sized 256×256 using 3,900 images from the Dresden Image Database [14]. Note that the training and testing patches were created from two separate sets of images. Each grayscale patch was edited using the three processing operations listed in Table 1 as well as all possible pairs of operations without excluding the redundant operations. This resulted in 13 different processing operations including the unaltered patches. Subsequently, each patch was JPEG compressed using a quality factor of 90.

Table 3 shows the results of our experiments using our constrained CNN and the SRM method to perform order of processing operations detection in re-compressed images. From Table 3, we can observe that one can improve CNN's performance using the deep features approach [11] by training an extremely randomized trees classifier. In all our experiemnts we used 800 trees to train the ERT classier. Noticeable, our ERT-based CNN can achieve 94.19% identification rate without redundant operations, which is particularly high. Also, we can notice that our ERT-based CNN approach outperforms the SRM method in both scenarios (i.e., redundant and non redundant operation scenarios).

Tables 4 and 5 contain the confusion matrices of respectively our ERT-based CNN and the SRM method with excluding the redundant processing operations. Due to space constraints, we only present the confusion matrix of our approach using the ERT classifier. From Table 4, one can notice that our approach can typically achieve higher than 91% detection accuracy with different types of image manipulations. Furthermore, we can observe that editing operations followed by median filtering are hard to detect. More specifically, the Guassian blur operation and resizing operation both followed by a median filtering (i.e., GB-MF and RS-MF) were detected with respectively 86.02% and 86% identification rates. This is mainly because the median filtering operation is forensically destructive to the former processing operations.

From Table 5, we can also notice that similarly to our approach it is challenging to detect image manipulations followed by median filtering using the SRM method. Additionally, we can observe that our ERT-based CNN approach outperforms the SRM method in detecting the different types of image manipulation typically by at least 1%, namely GB, RS, GB-MF, RS-MF and RS-GB. These results demonstrate the advantage of using our CNN-based approach over the SRM method. In what follows we present our results for the redundant processing operations scenario.

Tables 6 and 7 depict the confusion matrices of respectively our ERT-based CNN and the SRM method in redundant processing operations scenario. From Table 6, we can notice that our ERT-based CNN approach can detect the different types of image manipulations with an identification rate typically higher than 83%. Similarly to our previous experiments for non redundant processing operations scenario, it is challenging to detect image manipulations followed by median filtering. One can observe that it is also challenging to detect median filtering which gets significantly confused with the redundant median filtering operation (i.e., MF-MF). This is because median filtering is idempotent such that the composition of median filtering operations is equivalent to a single median filtering operation.

From Table 7, we can notice that the SRM method achieved lower identification rate in detecting different types of manipulations. Noticeably, our ERT-based CNN outperforms the SRM method in detecting the following operations: MF, MF-GB, MF-RS, GB-RS, RS-GB, and RS-RS. All taken together, these results demonstrate again the advantage of our approach over the SRM method in both scenarios (without and with redundant operations). In the following set of experiments, we show how one can significantly improve CNN's performance by using a larger scale of training dataset. It is worth mentioning that in these experiments we extracted the spatial rich model features (i.e., SRMQ1) through a multi-threaded (using eight threads) implementation which took 17 hours per 100,000 image patches. This makes the SRM feature extraction extremely challenging, if not infeasible, using a very large database.

Effect of training set size

In general, a CNN's performance is dependent on the size and quality of the training set [22, 7]. Therefore we created a large training dataset for both scenarios, i.e., without and with redundant processing operations. To conduct these experiments, we collected 14,800 images from the Dresden Image Database.

Table 8: Testing accuracy with two different training data sizes in JPEG re-compressed images (QF=90)

	without redundant op.		**with redundant op.**	
#. training patches	296,000	2,368,000	769,600	3,078,400
Accuracy	92.90%	96.38%	85.66%	95.46%

Table 9: Confusion matrix for identifying the order of processing operations using our constrained CNN in JPEG re-compressed images (QF=90) without redundant operations; Constrained CNN was trained using 2,368,000 training image patches.

		Predicted Class									
		UA	MF	GB	RS	MF-GB	GB-MF	MF-RS	RS-MF	GB-RS	RS-GB
True Class	UA	**99.85%**	0.00%	0.10%	0.06%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	MF	0.04%	**90.50%**	0.08%	0.04%	0.31%	3.63%	0.06%	5.27%	0.00%	0.08%
	GB	0.00%	0.04%	**96.63%**	0.02%	0.35%	0.00%	0.00%	0.00%	0.00%	2.96%
	RS	0.06%	0.00%	0.00%	**99.83%**	0.00%	0.00%	0.08%	0.02%	0.02%	0.00%
	MF-GB	0.00%	0.00%	0.02%	0.02%	**96.77%**	2.67%	0.02%	0.06%	0.00%	0.44%
	GB-MF	0.00%	0.21%	0.08%	0.02%	1.92%	**94.37%**	0.04%	3.29%	0.02%	0.06%
	MF-RS	0.00%	0.00%	0.00%	0.04%	0.00%	0.00%	**99.94%**	0.02%	0.00%	0.00%
	RS-MF	0.00%	1.06%	0.02%	0.04%	0.12%	8.52%	0.71%	**89.44%**	0.04%	0.06%
	GB-RS	0.00%	0.02%	0.00%	0.02%	0.06%	0.02%	0.96%	0.10%	**98.52%**	0.31%
	RS-GB	0.00%	0.00%	1.44%	0.02%	0.23%	0.02%	0.06%	0.12%	0.08%	**98.04%**

Next, we created 256×256 grayscale patches in the same manner we described above. Each training image patch was edited using the same types of operations we used for both scenarios in all the previous experiments. We then JPEG re-compressed every training patch in both training datasets using a quality factor of 90.

In total, we created 2,368,000 training patches for the non redundant operations scenario and 3,078,400 training patches for the redundant operations scenario. Subsequently, we trained our proposed CNN to perform order of processing operations detection using these large scale training datasets in the same manner we described in the experimental setup Section. We evaluated the performance of our proposed method using the same testing datasets that we created in the previous set of experiments. In order to assess the impact of the training dataset size on CNN's performance we reported the identification rates using only the softmax layer.

Table 8 depicts the results of our experiments for the non redundant and redundant operation scenarios. From Table 8, one can notice that the identification rate has significantly improved in both scenarios. Noticeably, we can observe that in the redundant operations scenario the identification rate has improved by 9.8% when we increased the number of training image patches from 769,600 to 3,078,400. These results demonstrate that the order of processing operations detection task is challenging and requires a significant large amount of training patches. These results also demonstrate that CNN's performance is dependent of the size of the training dataset.

Table 9 shows the confusion matrix of our approach in non redundant processing operations scenario. From Tables 9 and 4, we can notice that using a larger number of training image patches one can improve the identification rate typically for all the types of processing operations except for median filtering. Additionally, the identification rate for the processing operations followed by median filtering has significantly improved. Noticeably, we can achieve 8.35% higher identification rate for Gaussian blur followed by median filtering using larger amount of training data.

Table 10 contains the confusion matrix of our approach in redundant processing operations scenario. From Tables 10 and 6, we can notice that similarly to the non redundant operations scenario one can improve the identification rate typically for all the types of processing operations using a larger number of training image patches except for the following three operations: RS-GB, GB-GB, and GB-MF. Additionally, the identification rate for the processing operations followed by median filtering has significantly improved. Noticeably, we can achieve 51.53% higher identification rate for median filtering followed by median filtering using larger amount of training data. Taken all together, these results demonstrate again that the challenging order of processing operations detection task requires a large amount of training patches to have a better data representation for every type of image manipulation.

Experimental results summary

In our experiments, we investigated the ability of our CNN-based approach to forensically detect the order of processing operations in JPEG re-compressed images. First, we experimentally demonstrated that CNNs associated with the constrained convolutional layer are good candidates to extract low-level prediction error features and to determine the order in which a processing chain have been applied to an image. Specifically, when CNN is associated with the constrained convolutional layer it outperforms the other choices of CNN associated with different low-level feature extractors, namely the fixed HPF and the MFR feature extractor, by at least 12.27% identification rate in JPEG re-compressed images. These results suggest that the fixed low-level feature extractors may learn suboptimal features as opposed to our adaptive constrained convolutional layer which can lead to significantly better performance in JPEG re-compressed images.

Next, we compared our CNN-based approach to the SRM

Table 10: Confusion matrix for identifying the order of processing operations using our constrained CNN in JPEG re-compressed images (QF=90) with redundant operations; Constrained CNN was trained using 3,078,400 training image patches.

		Predicted Class												
		UA	MF	GB	RS	MF-MF	MF-GB	GB-MF	MF-RS	RS-MF	GB-GB	GB-RS	RS-GB	RS-RS
True Class	UA	**99.97%**	0.00%	0.00%	0.03%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	MF	0.03%	**87.19%**	0.00%	0.00%	4.81%	0.56%	2.59%	0.03%	4.53%	0.25%	0.00%	0.00%	0.00%
	GB	0.00%	0.00%	**93.00%**	0.00%	0.00%	0.06%	0.00%	0.00%	0.00%	4.53%	0.00%	2.41%	0.00%
	RS	0.03%	0.00%	0.00%	**99.78%**	0.00%	0.00%	0.00%	0.00%	0.03%	0.00%	0.03%	0.00%	0.12%
	MF-MF	0.00%	0.75%	0.00%	0.00%	**98.31%**	0.06%	0.38%	0.09%	0.41%	0.00%	0.00%	0.00%	0.00%
	MF-GB	0.00%	0.00%	0.00%	0.00%	0.44%	**97.28%**	0.69%	0.00%	0.00%	1.38%	0.03%	0.19%	0.00%
	GB-MF	0.00%	0.31%	0.06%	0.03%	1.69%	3.94%	**89.53%**	0.00%	4.38%	0.00%	0.03%	0.03%	0.00%
	MF-RS	0.00%	0.00%	0.00%	0.06%	0.00%	0.00%	0.00%	**99.69%**	0.09%	0.00%	0.03%	0.00%	0.12%
	RS-MF	0.00%	0.78%	0.03%	0.03%	0.81%	0.41%	6.81%	0.25%	**90.59%**	0.06%	0.09%	0.00%	0.12%
	GB-GB	0.00%	0.06%	0.09%	0.00%	0.00%	1.06%	0.00%	0.00%	0.03%	**94.97%**	0.09%	3.69%	0.00%
	GB-RS	0.00%	0.00%	0.00%	0.00%	0.03%	0.00%	0.00%	0.28%	0.06%	0.06%	**99.44%**	0.06%	0.06%
	RS-GB	0.00%	0.00%	0.31%	0.00%	0.00%	0.47%	0.00%	0.00%	0.00%	7.44%	0.41%	**91.34%**	0.03%
	RS-RS	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.03%	0.00%	0.00%	0.00%	0.00%	**99.97%**

method in two scenarios: (1) without redundant processing operations and (2) with redundant processing operations. Results of these experiments showed that our proposed ERT-based CNN can achieve higher identification rate in both scenarios. This demonstrates that our CNN-based data driven approach can capture order of manipulation traces that the hand-designed SRM features based method cannot detect. Furthermore, experiments showed that it is challenging to detect image manipulations when they are followed by median filtering for both approaches, i.e. ERT-based CNN and SRM. This was expected since the median filtering is commonly known for being forensically destructive. That is, many forensic fingerprints left in images by different types of image manipulation can potentially be hidden/erased by the median filtering operation.

In response to this, we trained our proposed CNN with a larger scale training dataset. Results of these experiments showed that one can significantly improve CNN's performance when using a larger scale of training dataset. Noticeably, we can achieve 9.8% higher identification rate in the redundant processing operations scenario when we increased the total number of training image patches from $769,600$ to $3,078,400$. Training the CNN with a larger amount of data has also significantly improved the detection rate of processing operations followed by median filtering. These experiments suggest that the order of processing operation task requires a large amount of training data to capture the manipulation fingerprints left by different types of single editing operations as well as chains of editing operations.

Conclusion

In this paper, we proposed a data-driven approach to performing order of processing operations detection in JPEG re-compressed images. Traditional approaches to order of processing operations detection rely on theoretical analysis of parametric models which may not be accurate enough. Instead, our proposed method is able to learn directly from data the cumulative effect induced by a sequence of processing operations. Specifically, we cast the order of processing detection as a classification problem where each sequence of manipulations corresponds to a new class of ordered chain of editing operations. To accomplish this, we used a constrained CNN, which employs a constrained convolutional layer, to directly extract from data low-level pixel relationships that capture the unique forensic fingerprints induced by each ordered chain of processing operations. Our CNN-based detector can be scalable and generic to detect editing chains with multiple types and number of editing operations.

We first evaluated the performance of our CNN-based detector with different low-level feature extractors commonly used in forensics. Next, we compared our constrained CNN based method to the spatial rich model approach in redundant and non redundant operation scenarios. Results of these experiments demonstrated that our proposed method outperforms the SRM in both scenarios (i.e., redundant and non redundant operations). Finally, we experimentally demonstrated that one can significantly improve CNN's performance at performing order of operations detection by using a larger amount of training image patches. Particularly, in redundant operations scenario we can achieve 9.8% higher detection rate using a significantly larger training dataset. This suggests that the order of processing operations detection task requires a large amount of data to train a CNN given the challenging scenarios that we considered in our experiments.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1553610. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] B. Bayar and M. C. Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Workshop on Information Hiding and Multimedia Security*, pages 5–10. ACM, 2016.

[2] B. Bayar and M. C. Stamm. Augmented convolutional feature maps for robust cnn-based camera model identification. In *The 2017 IEEE International Conference on Image Processing*. IEEE, 2017.

[3] B. Bayar and M. C. Stamm. Design principles of convolutional neu-

ral networks for multimedia forensics. In *International Symposium on Electronic Imaging*. IS&T, 2017.

[4] B. Bayar and M. C. Stamm. A generic approach towards image manipulation parameter estimation using convolutional neural networks. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2017.

[5] B. Bayar and M. C. Stamm. On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection. In *The 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2152–2156. IEEE, 2017.

[6] B. Bayar and M. C. Stamm. Towards open set camera model identification using a deep learning framework. In *The 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[7] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer, 2012.

[8] M. Boroumand and J. Fridrich. Scalable processing history detector for jpeg images. In *International Symposium on Electronic Imaging: Media Watermarking, Security, and Forensics*. IS&T, 2017.

[9] J. Chen, X. Kang, Y. Liu, and Z. J. Wang. Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, 22(11):1849–1853, Nov. 2015.

[10] V. Conotter, P. Comesana, and F. Pérez-González. Forensic detection of processing operator chains: Recovering the history of filtered jpeg images. *IEEE Transactions on Information Forensics and Security*, 10(11):2257–2269, 2015.

[11] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[12] J. Fridrich and J. Kodovskỳ. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.

[13] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[14] T. Gloe and R. Böhme. The dresden image database for benchmarking digital image forensics. *Journal of Digital Forensic Practice*, 3(2-4):150–159, 2010.

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

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[17] X. Kang, M. C. Stamm, A. Peng, and K. J. R. Liu. Robust median filtering forensics using an autoregressive model. *IEEE Trans. Information Forensics and Security,*, 8(9):1456–1468, Sept. 2013.

[18] M. Kirchner. Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In *Proceedings of the 10th ACM Workshop on Multimedia and Security*, MM&Sec '08, pages 11–20, New York, NY, USA, 2008. ACM.

[19] M. Kirchner and J. Fridrich. On detection of median filtering in digital images. In *IS&T/SPIE Electronic Imaging*, pages 754110–754110. International Society for Optics and Photonics, 2010.

[20] L. Pibre, P. Jérôme, D. Ienco, and M. Chaumont. Deep learning for steganalysis is better than a rich model with an ensemble classifier, and is natively robust to the cover source-mismatch. *arXiv preprint arXiv:1511.04855*, 2015.

[21] A. C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, Feb. 2005.

[22] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962, 2003.

[23] M. C. Stamm, X. Chu, and K. R. Liu. Forensically determining the order of signal processing operations. In *The 2013 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 162–167. IEEE, 2013.

[24] M. C. Stamm and K. J. R. Liu. Forensic detection of image manipulation using statistical intrinsic fingerprints. *IEEE Trans. on Information Forensics and Security*, 5(3):492 –506, 2010.

[25] M. C. Stamm, M. Wu, and K. J. R. Liu. Information forensics: An overview of the first decade. *IEEE Access*, 1:167–200, 2013.

[26] 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, 2016.

Author Biography

***Belhassen Bayar** received the B.S. degree in Electrical Engineering from the Ecole Nationale d'Ingénieurs de Tunis (ENIT), Tunisia, in 2011, and the MS degree in Electrical and Computer Engineering from Rowan University, New Jersey, in 2014. After graduating from ENIT, he worked as a Research Assistant at the University of Arkansas at Little Rock (UALR). In Fall 2014, he joined Drexel University, Pennsylvania, where he is currently a PhD candidate with the Department of Electrical and Computer Engineering. Bayar won the Best Paper Award at the IEEE International Workshop on Genomic Signal Processing and Statistics in 2013. In summer 2015 he interned at Samsung Reasearch America in Mountain View, California. His main research interests are in image forensics, machine learning and signal processing.*

***Matthew C. Stamm** received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Maryland, College Park in 2004, 2011, and 2012, respectively. He is an Assistant Professor with in the Department of Electrical and Computer Engineering at Drexel University. From 2004 to 2006, he was a Radar Systems Engineer with the Johns Hopkins University Applied Physics Laboratory. His research interests include multimedia forensics, signal processing, information security, and machine learning.*