Scalable Processing History Detector for JPEG Images

Mehdi Boroumand and Jessica Fridrich, Department of ECE, SUNY Binghamton, NY, USA, {mboroum1,fridrich}@binghamton.edu

Abstract

Knowing the history of global processing applied to an image can be very important for the forensic analyst to correctly establish the image pedigree, trustworthiness, and integrity. Global edits have been proposed in the past for "laundering" manipulated content because they can negatively affect the reliability of many forensic techniques. In this paper, we focus on the more difficult and less addressed case when the processed image is JPEG compressed. First, a bank of binary linear classifiers with rich media models are built to distinguish between unprocessed images and images subjected to a specific processing class. For better scalability, the detector is not built in the rich feature space but in the space of projections of features on the weight vectors of the linear classifiers. This decreases the computational complexity of the detector and, most importantly, allows estimation of the distribution of the projections by fitting a mutlivariate Gaussian model to each processing class to construct the final classifier as a maximum-likelihood detector. Well-fitting analytic models permit a more rigorous construction of the detector unachievable in the original high-dimensional rich feature space. Experiments on grayscale as well as color images with a range of JPEG quality factors and four processing classes are used to show the merit of the proposed methodology.

Motivation

Establishing the processing history of an image is important for an image analyst because forensic tools that attempt to determine image integrity and origin generally exhibit varying degree of sensitivity to non-malicious processing, such as tonal adjustment and filtering. Knowing the history of processing is also useful for establishing the chain of custody and provenance of digital evidence in legal cases and for intelligence gathering and interpretation to reveal deception attempts, such as "laundering" of manipulated images. Moreover, a processing detector could be applied on smaller image tiles to detect local inconsistencies due to content replacement (digital forgeries).

There is one obvious fundamental problem with processing history recovery and that is what constitutes an "unprocessed" image. All images acquired by a sensor are by definition processed either directly in the camera or in post-production in image editing software. In particular, all images are processed using non-linear tonal curves enhancing midtones to match the sensor sensitivity to light to that of the human eye. Also, different tone mapping may be applied to the image inside the camera based on whether the camera is set to "sunset mode", "night mode", or whether it is in "auto" mode. Niche processing, such as HDR photography, image stitching, focus bracketing, are other examples of more advanced operations that can be applied to images inside the camera. On the other hand, regardless of the manufacturer, all cameras by definition strive to produce natural looking output. Thus, detection of global image processing operations should be understood as detection of deviations from typical camera outputs. Taking the example of tonal adjustment, if a raw sensor output was not adjusted, its midtones would be too dark, which should be detected as unusual.

A large bulk of prior art focuses on specific processing, such as median filtering [6, 7, 20, 38], tonal adjustment [31, 32, 2, 12, 37], resizing [10, 13, 15, 18, 21, 19, 23, 26, 27, 28, 34, 36], and multiple JPEG compression [3, 4, 25, 29]. Detection of filtering was investigated in [33]. The problem of recovering the order of processing has been studied from the information-theoretical perspective in [9].

Most methods work well only for uncompressed images and their accuracy may drop significantly with JPEG compression. In this paper, we focus on the case when the processed image is stored in the JPEG format as this is by far the most common case in practice. In the presence of JPEG compression, many forensic tools based on pixel descriptors start breaking. For example, artifacts of color interpolation or lens-distortion correction may be severely suppressed. This makes forensics of JPEG images challenging but not impossible because the DCT coefficients will still exhibit anomalous statistical properties characteristic of their unique pedigree [9]. JPEG compression makes the problem harder because a substantial component of the fine-grain structure often leveraged for forensic purposes is significantly suppressed or distorted. The fact that JPEG compression depends on numerous parameters, such as luminance and chrominance quantization matrices and color subsampling, further complicates practical implementation of the processing detector. On the other hand, lossy compression has a tendency to equalize differences between sources and makes approaches that require supervised learning more suitable for practice due to smaller source mismatch.

Our goal is to devise a technique that will scale well w.r.t. the number of different processing classes and be robust to differences between sources and processing types. We desire a low computational complexity of testing, guaranteed error rates, and easy expandability. The proposed method utilizes rich image descriptors originally proposed for steganalysis. Forensic techniques utilizing repurposed steganalysis features were previously successfully used for detection of manipulated regions in, e.g., [35] and the references therein. A detector of processing that seems to work very well on uncompressed images was recently proposed based on a compactified spatial rich model in [22].

Motivated by scalability and computational efficiency, the proposed processing history detector is not built directly in the high-dimensional rich feature space but in a space with a much lower dimensionality formed by projections of features on weight vectors of linear classifiers trained to distinguish unprocessed images from images processed by a class of closely related operations. In this detection space, it becomes possible to estimate the statistical distribution of each processing class and build the processing classifier as a maximum likelihood detector.

In the next section, we describe the processing classes and their diversification used for building the detector and experiments. The detector architecture is explained in section Detector Architecture. The setup of experiments appears in section Setup of Experiments. The results of all experiments and their interpretation are in section Results. The paper is concluded in the last section.

Processing classes

Our goal is to identify which type of global image processing was applied to an image after the processed image has been JPEG compressed. Since each processing typically has parameters that control its strength and other properties, we group similar operations into classes to avoid excessive granularization of the detector. Besides, it may not be possible or desirable to reliably distinguish between two types of processing that are slightly different.

We work with $k = 4$ processing classes $C_1, \ldots, C_k$, that are typically applied to digital images and we added class C_0 containing unprocessed images. Each class has been diversified to include a spectrum of operations of similar character. Below, we describe the classes and their diversification. Note that each class contains exactly eight subclasses. In the description, we include the corresponding Matlab command in typewriter font whenever the processing was executed in Matlab. The number in square brackets is the number of subclasses of each type.

Low pass filtering

Low pass filtering is used to soften an image and to remove aliasing. It could also be used maliciously to blur crucial details in an image, such as a face or a license plate. We distinguish between low-pass filtering and denoising because denoising tries to preserve the sharpness of edges and textures while, what we understand by low-pass filtering is applied uniformly to the image. This class has been diversified in the following manner.

- [3] Filtering with an averaging kernel 3×3, 5×5, and 7×7, `imfilter(`X`,fspecial('average',3),'symmetric')`
- [4] Filtering with a Gaussian kernel 3×3 and 5×5 with $\sigma \in \{0.5, 1\}$, `imfilter(`X`,fspecial('gaussian',3,`σ`), 'symmetric')`
- [1] Filtering with the abs. value of the KB filter [5]:

$$\frac{1}{4}\begin{pmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{pmatrix}. \quad (1)$$

Denoising

This is one of the most commonly executed operations as it improves the visual quality. It is often applied to images taken with a long exposure and/or high ISO, such as images taken under low light conditions or zoomed images taken without some form of optical/electronic/mechanical stabilization.

- [3] Adobe Lightroom (ver. 5.7) luminance denoising with strength 20, 30, 50
- [2] Wiener filter denoising with default variance and 3×3 and 5×5 neighborhood, `wiener2(`X`,[3 3])`
- [3] Daubechies 8-tap wavelet denoising with $\sigma = 8, 10, 12$ [24] available from `http://dde.binghamton.edu/download/camera_fingerprint/`

High-pass filtering

Sharpening is commonly done to enhance detail in the image prior to printing or to achieve an artistic effect.

- [7] Sharpening with a Gaussian kernel G_σ of radius σ and amount α: $X \to X + \alpha(X - G_\sigma(X))$ with parameters $(\sigma, \alpha) \in \{(1, 0.8), (1, 1.5), (1.5, 1.5), (2, 1.5), (1, 2), (1.5, 2), (2, 2)\}$. `imsharpen(`$X$`,'Radius',`$\sigma$`,'Amount',`$\alpha$`)`
- [1] Unsharp masking with kernel

$$\frac{1}{3}\begin{pmatrix} -1 & -1 & -1 \\ -1 & 11 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad (2)$$

`imfilter(`X`,fspecial('unsharp',0.5), 'symmetric')`

Tonal adjustment

This class includes point transformations of pixel intensities. On the most general level, it is captured with a mapping $\tau : [0, 1] \to [0, 1]$, when scaling the pixel values to the unit interval. The choice of $\tau(x) = x^\gamma$ is gamma adjustment, linear $\tau(x)$ corresponds to brightness / contrast adjustment, etc. These operations are rather common and are applied by photographers to bring out detail in dark and bright regions. In Adobe Lightroom 5.7, they include the sliders "highlights", "shadows", "whites", "blacks", "exposure", "contrast", and user-defined tone curves.

- [4] Contrast adjustment: p% of the darkest (and brightest) pixels are mapped to 0 and 1 while the rest are stretched to [0,1], $p \in \{1, 2, 4, 6\}$, `imadjust(`X`,stretchlim(`$p/100$`),[])`
- [3] Contrast adjustment combined with gamma correction: after contrast adjustment with p, the luminance is processed with $x \to x^\gamma$ $(p, \gamma) \in \{(2, 0.8), (4, 0.6), (6, 1.2)\}$, `imadjust(`$X$`,stretchlim(`$p/100$`),[],`$\gamma$`)`
- [1] Histogram equalization, `histeq(`X`)`

We now make a few remarks regarding our choices. We do not consider multiple compression as a processing class since multiple JPEG compression has been extensively studied elsewhere. While there has been excellent prior

LPF	avg3	avg5	avg7	Gau3,0.5	Gau3,1	Gau5,0.5	Gau5,1	|KB|
	54.9	126	167	7.25	40.2	7.47	56.4	35.2
DEN	LR20	LR30	LR50	Wie3	Wie5	Wav8	Wav10	Wav12
	16.3	19.4	22.0	17.7	39.4	15.1	22.5	29.1
HPF	Imsh	1, 1.5	1.5, 1.5	2, 1.5	1, 2	1.5, 2	2, 2	Unsh
	31.4	106	157	198	165	254	336	319
TON	Con1%	Con2%	Con4%	Con6%	2%+0.8	4%+0.6	6%+1.2	HistEq.
	1631	1800	2390	3113	2633	4604	2618	4949

Table 1. MSE of all considered processing subclasses.

work done aimed at detection of tone adjustment based on histogram gaps and spikes [31, 32], these techniques are rather fragile when the processed image is JPEG compressed or filtered. We also want to point out that some of the considered processing operations are fairly gentle in the sense that they modify the image only very slightly, such as Gaussian blurring with $\sigma = 0.5$.

To give the reader a rough idea regarding the strength of the processing classes, Table 1 shows the mean square error (MSE) of each processing subclass.

Detector architecture

In this section, we describe the detectors used to determine the processing history of a JPEG image. First, we deal with a detector trained to recognize a single processing operation. It is then generalized to detect an entire processing chain. The basic building blocks of all detectors considered in this paper are binary classifiers trained on one half of the available dataset, designed to distinguish between unprocessed images from class $\mathcal{C}_0$ and each class (or processing chain). Since the training is an important part of the detector design, we also describe how the training and testing sets were prepared.

Detecting single processing

Starting with a dataset $\mathcal{D}$ of N_b never compressed images, they were all processed with operations from classes $\mathcal{C}_i$, $i \in \{1,\dots,k\}$ to create k more datasets $\mathcal{D}_i$ of the same cardinality as $\mathcal{D}$. Since each class contains 8 subclasses of related operations, 1/8 of images in $\mathcal{D}_i$ selected at random were processed with each subclass. For example, for the denoising class (Section Processing Classes), 1/8 of images from $\mathcal{D}$ were processed with Lightroom with denoising strength 20, another 1/8 with strength 30, etc. Their union forms the dataset $\mathcal{D}_i$ containing the same number of images as $\mathcal{D}$. Finally, all images from all datasets were JPEG compressed with the same compression parameters (quantization matrices and color subsampling). For simplicity, we will denote the datasets of compressed images with the same symbols.

The training and testing datasets were random divisions of equal size obtained with a random permutation π of indices $\{1,\dots,N_b\}$. Next, k binary classifiers $\phi^{(1)},\dots,\phi^{(k)}$ were trained for all k pairs of datasets $\mathcal{C}_0$,$\mathcal{C}_i$ on images with indices $\pi(1),\dots,\pi(N_b/2)$, the training part of the datasets. To be precise, to train the ith classifier $\phi^{(i)}$, the TRN set of images consisted of unprocessed images with indices $\pi(1),\dots,\pi(N_b/2)$ and images processed using class $\mathcal{C}_i$ with indices $\pi(1),\dots,\pi(N_b/2)$. This way, we made sure that no image used for training appears in any other form in any of the testing sets formed by images with indices $\pi(N_b/2+1),\dots,\pi(N_b)$.

Formally, each detector $\phi^{(i)}$ is described by its weight vector $\mathbf{w}^{(i)} \in \mathbb{R}^d$, where d is the feature dimensionality. Each weight vector was normalized to a unit L_2 norm. The soft output of detector $\phi^{(i)}$ on image $\mathbf{x}$ is obtained as the dot product (projection) $\mathbf{w}^{(i)} \cdot f(\mathbf{x})$, where $f(\mathbf{x})$ is the feature of image $\mathbf{x}$. Thus, with k detectors, we obtain a k-dimensional vector of projections $\mathbf{p}(\mathbf{x}) = \Big(\mathbf{w}^{(1)} \cdot f(\mathbf{x}),\dots,\mathbf{w}^{(k)} \cdot f(\mathbf{x})\Big) \in \mathbb{R}^k$. Note that $k \ll d$.

The k-dimensional vectors for all images in the training set ($N_b/2 \times (k+1)$ images total) form $k+1$ clusters corresponding to $k+1$ classes. To obtain a multi-class detector with probabilistic output, we model the distribution of $\mathbf{p}(\mathbf{x})$ in each cluster using a multivariate Gaussian (MVG) distribution with means and covariances computed as sample expectations, $l = 0,\dots,k$:

$$\overline{\mathbf{p}}^{(l)} = \mathrm{E}_{\mathbf{x}\in\mathcal{C}_l}[\mathbf{p}(\mathbf{x})] \tag{3}$$

$$(\mathbf{C}^{(l)})_{ij} = \mathrm{E}_{\mathbf{x}\in\mathcal{C}_l}\left[\left(p_i(\mathbf{x}) - \overline{\mathbf{p}}_i^{(l)}\right)\left(p_j(\mathbf{x}) - \overline{\mathbf{p}}_j^{(l)}\right)\right]. \tag{4}$$

Given a projection vector $\mathbf{p}(\mathbf{x})$ of image $\mathbf{x}$, the maximum likelihood detector assigns it to class $\hat{l}$ such that

$$\hat{l} = \arg\max_l \varphi(\mathbf{p}(\mathbf{x}); \overline{\mathbf{p}}^{(l)}, \mathbf{C}^{(l)}), \tag{5}$$

where $\varphi(\mathbf{p}(\mathbf{x}); \overline{\mathbf{p}}^{(l)}, \mathbf{C}^{(l)})$ is the multivariate Gaussian density with mean $\overline{\mathbf{p}}^{(l)} \in \mathbb{R}^k$, covariance matrix $\mathbf{C}^{(l)} \in \mathbb{R}^{k\times k}$ evaluated at $\mathbf{p}(\mathbf{x})$. Note that if a priori class probabilities are available, one could also use a MAP detector.

Detecting processing chain

In this section, we generalize the detector to be able to detect an entire chain of processing operations, including their order. The most straightforward option is to keep the ML (MAP) principle and enlarge the number of processing classes indexed by the processing chain. In particular, if we consider chains of up to m operations, we will have

$$c_m = \sum_{i=0}^{m} k^i = \frac{k^{m+1}-1}{k-1} \tag{6}$$

processing chains of length m and then estimate the parameters of c_m k-dimensional MVGs.

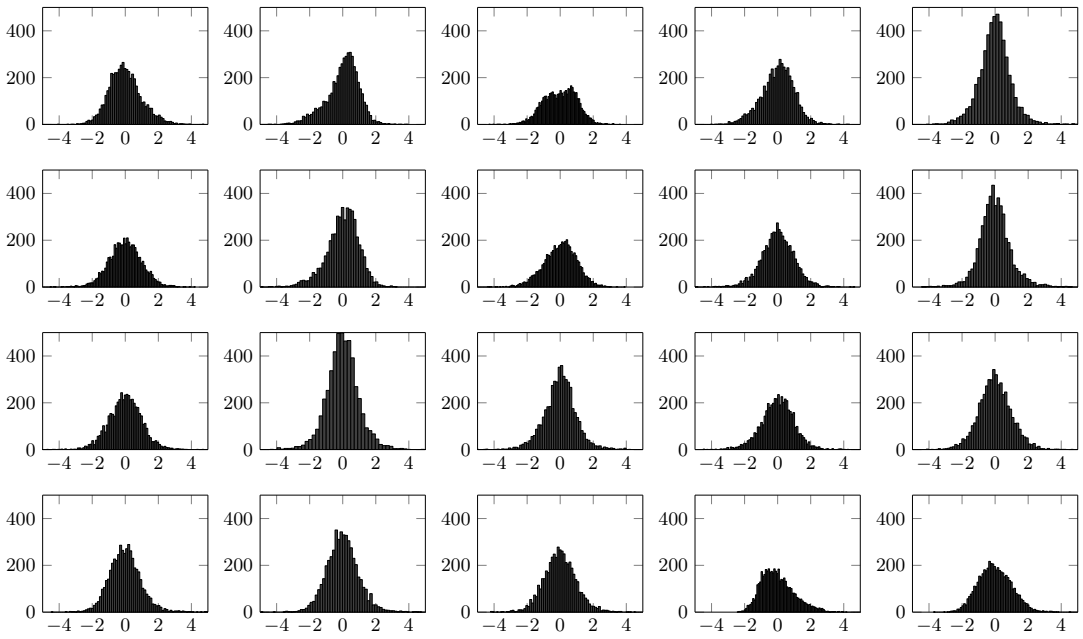

Figure 1. *Histogram of projections of images from five classes (columns) onto $k = 4$ weight vectors after decorrelation and normalization to zero mean and unit variance. Rows correspond to coordinates after decorrelation, columns 1–5 to classes U, L, D, H, T. Dataset BOSSbase 1.01, SRMQ1 features, JPEG quality factor 85.*

Training a detector of a processing chain naturally requires more training examples. In this paper, we only tested this detector for $m = 2$ with $c_2 = 21$ chains. Starting with the dataset $\mathcal{D}$, for processing chain l_1, l_2, $0 \le l_1, l_2 \le k$, every image was processed with one randomly selected subclass from class $\mathcal{C}_{l_1}$ (with 8 subtypes) and then with a randomly selected subclass from $\mathcal{C}_{l_2}$. Thus, on average the training set contained $N_b/2/64$ images processed with the same pair of subclasses in the same order.

This approach does not scale well with respect to the chain length m. A possible alternative is to keep the number of classes at $k+1$ and threshold the likelihoods. This would not, however, allow us to recover the processing order or tell when an image was processed with two or more operations from the same class.

In general, both the single-processing detector and the detector of the chain avoid modeling the statistical distribution of high-dimensional descriptors, which is generally infeasible due to the curse of dimensionality. Estimating the MVGs, on the other hand, is feasible with typical datasets or order $10^4 - 10^7$. Expanding the single-processing detector to consider an additional class can be done rather efficiently as one only needs to train one more binary linear classifier and reestimate k MVG distributions in k dimensions.

Setup of experiments

In this section, we describe the specifics of our detectors, including the features, and the datasets. We also validate our modeling assumptions.

Feature sets and classifier

For grayscale images, we investigated two types of descriptors previously developed for steganalysis: the 12,753-dimensional spatial rich model with quantization step 1 (SRMQ1) [14] and the 17,000 dimensional Gabor Filter Residual (GFR) features [30, 11]. The SRM has previously found numerous applications in forensics and is thus a natural choice. The GFR features are an example of so-called JPEG-phase-aware features [16, 17, 30, 11] that are known to be very effective for detection of modern JPEG steganography. For color images, we used the 18,157 dimensional spatio-color rich model (SCRM) because of its ability to capture dependencies among color channels.

Since our construction calls for linear classifiers, we selected the regularized linear discriminant implemented using the LSMR optimizer [8] for the detector construction. The advantage of this particular classifier is its good performance and low computational complexity when used with high-dimensional features. The detector performance will be evaluated with confusion matrices and quantities extracted from them.

Datasets

Experiments with grayscale images were carried out on BOSSbase 1.01 [1] containing $N_b = 10,000$ images originally taken in the RAW format by seven different cameras, converted to 24-bit color images in 'dcraw', converted to 8-bit grayscale, downsampled so that the smaller dimension was 512 using the Lanczos resampling algorithm with antialiasing turned OFF, and centrally cropped to the final size of 512×512 pixels. Experiments with color images were executed on the same dataset processed in the same

GFR (dim 17,000)

$l \setminus \hat{l}$	U	L	D	H	T
U	3925	191	241	70	573
L	317	4251	392	13	27
D	473	188	4193	13	133
H	151	8	8	4508	325
T	1042	28	51	245	3634

SRMQ1 (dim 12,753)

$l \setminus \hat{l}$	U	L	D	H	T
U	4352	215	180	58	195
L	262	4500	190	10	38
D	311	227	4402	11	49
H	132	25	3	4674	166
T	355	20	9	244	4372

Table 2. Confusion matrix showing the number of cases class l was detected as class $\hat{l}$ using the ML detector (5) with binary classifiers implemented with GFR and SRMQ1 features. Dataset BOSSbase 1.01, JPEG quality factor 85.

GFR (dim 17,000)

$l \setminus \hat{l}$	1	2	3	4	5	6	7	8
LPF	avg3	avg5	avg7	Gau3,0.5	Gau3,1	Gau5,0.5	Gau5,1	|KB|
	600	**559**	**493**	**416**	**603**	**397**	**584**	**599**
DEN	LR20	LR30	LR50	Wie3	Wie5	Wav8	Wav10	Wav12
	280	**416**	**528**	**530**	**611**	**602**	**610**	**615**
HPF	Imsh	1, 1.5	1.5, 1.5	2, 1.5	1, 2	1.5, 2	2, 2	Unsh
	413	**578**	**585**	**574**	**596**	**605**	**562**	**567**
TON	Con1%	Con2%	Con4%	Con6%	2%+0.8	4%+0.6	6%+1.2	HistEq.
	329	**381**	**480**	**506**	**371**	**475**	**525**	**567**

SRMQ1 (dim 12,753)

$l \setminus \hat{l}$	1	2	3	4	5	6	7	8
LPF	avg3	avg5	avg7	Gau3,0.5	Gau3,1	Gau5,0.5	Gau5,1	|KB|
	612	**601**	**592**	**446**	**610**	**436**	**598**	**605**
DEN	LR20	LR30	LR50	Wie3	Wie5	Wav8	Wav10	Wav12
	381	**529**	**587**	**502**	**603**	**589**	**603**	**608**
HPF	Imsh	1, 1.5	1.5, 1.5	2, 1.5	1, 2	1.5, 2	2, 2	Unsh
	458	**596**	**597**	**598**	**606**	**616**	**604**	**599**
TON	Con1%	Con2%	Con4%	Con6%	2%+0.8	4%+0.6	6%+1.2	HistEq.
	389	**513**	**589**	**575**	**539**	**572**	**579**	**616**

Table 3. Detection of single processing by subclasses. Top: GFR, Bottom: SRMQ1. Dataset BOSSbase 1.01, quality factor 85. The detection rates are in boldface to make the reading easier.

GFR

	U	L	D	H	T	LL	LD	LH	LT	DL	DD	DH	DT	HL	HD	HH	HT	TL	TD	TS	TT
U	**184**	17	23	3	45	2	29	16	13	1	15	10	11	6	33	0	0	9	22	0	20
L	13	**203**	5	0	3	**203**	**29**	**85**	**127**	*136*	2	2	14	*151*	10	0	0	*169*	16	0	0
D	8	13	**197**	1	2	33	*170*	8	2	**101**	**216**	**84**	**98**	7	*145*	1	0	3	*186*	0	4
H	6	3	3	**218**	5	0	2	*113*	13	0	0	*119*	17	**73**	**36**	**226**	**199**	8	2	*168*	17
T	27	2	10	16	**183**	0	8	16	*83*	0	5	23	*98*	1	14	11	*39*	**22**	**12**	**70**	**197**

SRMQ1

	U	L	D	H	T	LL	LD	LH	LT	DL	DD	DH	DT	HL	HD	HH	HT	TL	TD	TS	TT
U	**206**	11	11	4	24	1	17	12	5	0	9	3	4	4	16	0	0	2	16	0	5
L	9	**218**	9	1	0	**218**	**33**	**100**	**53**	*155*	8	9	3	*160*	7	0	0	*154*	10	0	0
D	9	7	**213**	1	2	17	*183*	5	2	**83**	**218**	**55**	**43**	9	**179**	0	0	3	*177*	0	1
H	5	1	2	**221**	12	0	2	*114*	20	0	0	*142*	18	**64**	*25*	**235**	**202**	9	2	*205*	12
T	9	1	3	11	**200**	2	3	7	*158*	0	3	29	*170*	1	11	3	*36*	**70**	**33**	**33**	**220**

Table 4. Detecting doubly processed images with a single-processing detector. Top: GFR, Bottom: SRMQ1. BOSSbase, JPEG quality factor 85.

manner with the exception of skipping the conversion from RGB to grayscale. We will use the acronym BOSSColor for this source.

MVG model verification

The maximum likelihood detector uses MVG distributions estimated from data. To verify the modeling assumptions, we decorrelated the projections of images from class l using the Mahalanobis transformation executed by first subtracting the sample mean $\overline{\mathbf{p}}^{(l)}$ (3) and then multiplying the projections by the square root of the inverse covariance matrix $\mathbf{C}^{(l)}$ (4) (implemented using Matlab command `'sqrtm.m'`). Under the MVG assumption, this should make the distributions of all four transformed coordinates follow $\mathcal{N}(0,1)$. The histograms of projections of each coordinate are shown as rows in Figure 1. The coordinates (rows) correspond to projections on weight vectors of L, D, H, T classes while the columns correspond to classes U, L, D, H, T. The projections mostly follow a unimodal distribution with thin tails that seems to be reasonably approximated by a Gaussian.

Results

This section contains the results of all experiments and their discussion. First, we carry out an extensive pilot experiment on grayscale images with quality factor 85. The purpose is to assess which feature set is more suitable for the task and evaluate the ability of the processing chain detector to identify both processing operations as well as their order. The results of the single processing detector across three JPEG quality factors are reported using confusion matrices at the end of this section. All experiments with color images appear in the second section.

Grayscale images

Table 2 shows the confusion table for the single processing detector implemented with both the GFR and SRMQ1 features. The detection accuracy broken up for each subclass is shown in Table 3. The maximum value in Table 2 is 5,000 while in tables by subclasses it is $5,000/8 = 625$. As expected, the accuracy across subclasses for a given class correlates with the MSE (Table 1). The lowest detection rates correspond to the weakest operations and vice versa.

In the next experiment, we tested the single-processing detector on doubly processed images to see if it can correctly detect at least one of the operations and whether the order matters. For this experiment, the 5,000 test images were divided into 21 disjoint groups of 238 ($21 \times 238 \approx 5,000$), each group corresponding to one pair out of 21 possibilities how to apply two of the four classes. Note that the 21 cases include singly processed images (columns 2–5) as well as the cases when the same processing was applied twice, LL, DD, HH, and TT. The columns in Table 4 show the number of cases when a specific chain (shown in the top row) was detected as a single processing. For example, the column LD tells us that out of the 238 images processed with LD (L first, D second), 29 were detected as unprocessed, 29 as L, 170 as D, 2 as H, and 8 as T. The bold face number in each column corresponds to the first processing while the italic to the second processing. Single processing as well as repetitive processing with the same class only have the bold face value. The maximum value in each bin is $5,000/21 = 238$. Overall, the single processing detector does a very good job in mostly detecting the two applied operations. It is also apparent that the second operation is more likely to be detected than the first one.

The next experiment for quality factor 85 and grayscale images concerns a detector designed to detect a chain of $m = 2$ operations. The training of the $c_2 = 21$ binary classifiers was executed as described in Section with the final detector built using the maximum-likelihood principle. Instead of showing the complete 21×21 confusion matrix, we summarize the most important findings. We note that certain operations, such as histogram equalization, when applied twice lead to the same result as when applied once. Also, some processing chains will fundamentally appear as single processing, such as contrast enhancement with p followed by contrast enhancement with $q < p$. These cases have been taken into account when interpreting the results below.

- In 41.5% cases, both processing operations were correctly identified, including their order. A randomly guessing detector would be successful in $1/21 = 4.76\%$ cases.
- If processing $\mathrm{X} \in \{\mathrm{L}, \mathrm{D}, \mathrm{H}, \mathrm{T}\}$ is part of the chain, it is detected as one of the processing operations in 85%, 81%, 74%, and 68% of cases, respectively.
- If two processing operations are detected, the second is almost twice as likely to be correct than the first (2052 vs. 1228 cases).
- The last processing detected is correctly detected as last in 71% cases.
- The first processing detected is correctly detected as first in 50% cases.

Overall, one can say that detection of a processing chain of length 2 is less reliable than detecting a single processing. On the other hand, upon inspecting the missed detections, the cases that lead to errors are mostly when a strong processing is combined with a subtle one. Missing a subtle processing is likely going to be less of a problem for the analyst as weak operations will have correspondingly smaller impact on the reliability of other forensic methods.

Finally, in Figures 2 and 3 we provide a graphical representation of the cluster centers corresponding to each processing class (processing chain) obtained by reducing the four-dimensional detection space to two dimensions using a dimensionality reduction technique called multidimensional scaling (MDS). Especially Figure 3 is informative as one can infer that the T and TT classes as well as D and DD appear rather close to each other. This is not the case for H and HH. The chains HD and DH have a tendency to cancel each other, which is confirmed in the figure as they are among the closest to U. Comparing the centers between both figures, we can see that the denoising class got broken up into subclusters of chains with D as the

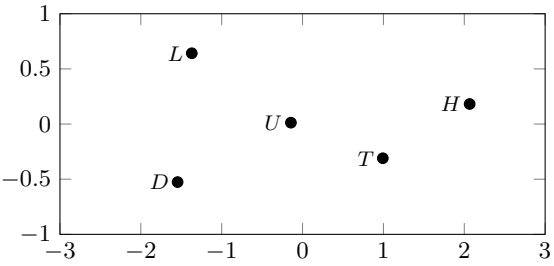

Figure 2. *The detection space reduced from dimension 4 to 2 using MDS, single processing detector, BOSSbase, quality factor 85, SRMQ1 features.*

second applied operation. This is not the case, however, for the other operations.

Table 5 shows the confusion matrices across three JPEG quality factors for the SRMQ1 feature set. As expected, the detection accuracy increases with higher quality JPEG and decreases with lower JPEG quality. The detector has a good accuracy even for the smallest tested quality factor of 75.

Color images

The last experiment reported in this paper corresponds to the case of color images from BOSSColor. The four processing classes were prepared and diversified in the same fashion as for the grayscale images. Experiments with a single-processing detector were executed on three quality factor, 75, 85, and 95, while the double-processing detector was tested only for quality factor 85.

Contrary to our expectations, the color information does not improve the reliability of the processing detector. The accuracy of the single processing detector (correctly detecting a single processed image) and its false alarm (detecting an unprocessed image as processed) across quality factors and grayscale as well as color images is contrasted in Table 6. The classification performance of the single processing detector applied to doubly processed images is shown in Table 7, while Table 8 shows the confusion matrices for three JPEG quality factors.

The performance of the double processing detector is only summarized in a similar fashion as for grayscale images.

- The double processing detector correctly detected the entire chain of two operations in 40.7%.
- If processing $X \in \{L, D, H, T\}$ is part of the chain, it is detected as one of the processing operations in 84%, 80%, 74%, and 71% of cases, respectively.
- If two processing operations are detected, the second is almost twice as likely to be correct than the first (1990 vs. 1247 cases).
- The last processing detected is correctly detected as last in 68% cases.
- The first processing detected is correctly detected as first in 49% cases.

Conclusions

In this paper, we proposed a novel detector of global processing applied to an image prior to JPEG compression. For better scalability, we construct the detector using the maximum likelihood principle in a space of projections on discrimination vectors between the unprocessed class and all other processing classes obtained from linear classifiers trained in a rich feature space. Because the ML is constructed in the space of projections, whose dimensionality is the number of processing classes (or processing chains for detection of an entire chain of operations), it is feasible to use parametric models. The merit of the approach is demonstrated on grayscale and color images for a range of quality factors. Four processing classes are investigated – low-pass filtering, high-pass filtering, denoising, and tonal adjustment. The detector for grayscale images was implemented using the SRMQ1 model while for color images, the SCRMQ1 model was used. Since the color part of the SCRMQ1 model, the Color Rich Model (CRM), did not improve the performance in any significant manner, as part of our future effort, we plan to investigate novel forms of the rich model for color images that are better suited for decompressed JPEGs.

This work is a mere initial direction. For practical applications, we need to address the diversity of JPEG quantization tables. This can be achieved by constructing detectors for each quality factor. JPEG images with a non-standard quantization table will be classified with a classifier trained on the closest standard table.

It is also anticipated that further performance drop / complexity increase will be experienced when trying to detect a longer chain of operations. In this case, perhaps, it would be reasonable to limit our ambitions to detecting only the strongest operations.

We would also like to point out some natural as well as fundamental limitations. For example, it may be rather challenging to distinguish an out-of-focus image of a flat scene from a low-pass filtered (blurred) image. Strong operations may overpower (neutralize the impact of) others, e.g., sharpening followed by aggressive low-pass filtering. Also, the order of operations that commute cannot be established, e.g., contrast and linear filtering.

In the future, we plan to study a different setup in which the pipeline starts with a decompressed JPEG, which is then decompressed, processed, resized, and then JPEG compressed to simulate what happens when imagery is uploaded to Facebook (resizing and compression is also used for image laundering to prevent forensic methods from being applicable). For this case, we plan to strengthen the tested processing as it is unlikely that subtle processing will impact the images enough for a reliable detection.

Acknowledgments

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) 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

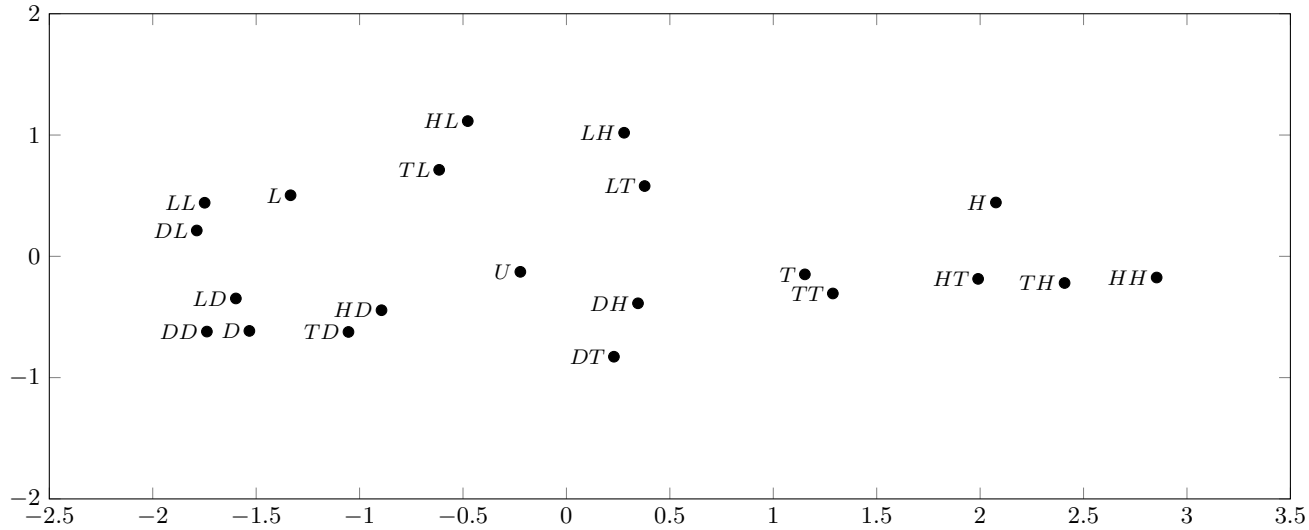

Figure 3. *The detection space reduced from dimension 4 to 2 using MDS, double processing detector, BOSSbase, quality factor 85, SRMQ1 features.*

QF 75						QF 85						QF 95					
$l \setminus \hat{l}$	U	L	D	H	T	$l \setminus \hat{l}$	U	L	D	H	T	$l \setminus \hat{l}$	U	L	D	H	T
U	4158	263	263	56	260	U	4352	215	180	58	195	U	4670	80	94	23	133
L	382	4215	288	66	49	L	262	4500	190	10	38	L	112	4649	224	4	11
D	321	280	4319	13	67	D	311	227	4402	11	49	D	176	187	4500	13	124
H	171	31	6	4649	143	H	132	25	3	4674	166	H	27	9	3	4786	175
T	376	25	25	230	4344	T	355	20	9	244	4372	T	289	13	19	289	4390

Table 5. Confusion matrices for the single-processing detector across JPEG quality. Feature set SRMQ1, BOSSbase.

	False alarm			Correct detection		
QF	75	85	95	75	85	95
BOSSbase	.168	.130	.066	.876	.897	.916
BOSSColor	.167	.129	.097	.870	.898	.915

Table 6. False alarm and correct processing detection probability of the ML detector on grayscale and color images.

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 and Air Force Research Laboratory (AFRL) or the U.S. Government.

References

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

[2] S. Battiato, G. Messina, and D. Strano. Chain of evidence generation for contrast enhancement in digital image forensics. In R. Creutzburg and D. Akopian, editors, *Multimedia on Mobile Devices*, volume 7542 of *Proceedings of SPIE*, page 75420E. SPIE, 2010.

[3] T. Bianchi and A. Piva. Analysis of non-aligned double JPEG artifacts for the localization of image forgeries. In *IEEE International Workshop on Information Forensics and Security, WIFS 2011*, 2011.

[4] T. Bianchi and A. Piva. Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, 2012.

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

[6] Gang Cao, Yao Zhao, Rongrong Ni, Lifang Yu, and Huawei Tian. Forensic detection of median filtering in digital images. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 89–94, July 2010.

[7] C. Chen and J. Ni. Median filtering detection using edge based prediction matrix. In Y. Q. Shi, H.-J. Kim, and F. Pérez-González, editors, *Digital Forensics and Watermarking, 10th International Workshop, IWDW 2011*, number 7128 in Lecture Notes in Computer Science, pages 361–375, Berlin, Heidelberg, 2011. Springer-Verlag.

[8] 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.

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

	SCRMQ1																				
	U	L	D	H	T	LL	LD	LH	LT	DL	DD	DH	DT	HL	HD	HH	HT	TL	TD	TS	TT
U	**205**	13	19	4	33	0	25	9	1	2	11	3	4	5	16	0	0	0	18	2	5
L	10	**206**	11	1	0	**211**	**47**	**75**	**86**	*167*	6	6	4	*156*	19	1	0	*150*	12	1	0
D	8	14	**203**	0	0	24	*158*	3	8	**69**	**219**	**48**	**91**	3	**165**	0	0	4	*166*	0	3
H	4	1	2	**227**	9	0	1	*140*	17	0	0	*128*	2	**66**	*26*	**235**	**206**	7	1	*201*	6
T	11	4	3	6	**196**	3	7	11	*126*	0	2	53	*137*	8	12	2	*32*	**77**	**41**	**34**	**224**

Table 7. Detecting doubly processed images with a single-processing detector. SCRMQ, BOSSColor, JPEG quality factor 85.

	QF 75					QF 85					QF 95				
$l \setminus \hat{l}$	U	L	D	H	T	U	L	D	H	T	U	L	D	H	T
U	4166	276	249	61	248	4353	203	168	39	237	4517	68	134	16	265
L	379	4181	321	58	61	293	4382	248	26	51	126	4514	282	45	33
D	388	288	4227	18	79	325	206	4397	5	67	204	234	4493	5	64
H	158	18	7	4671	146	99	11	2	4797	91	12	7	2	4828	151
T	445	22	32	182	4319	441	30	20	122	4387	383	23	36	98	4460

Table 8. Confusion matrices for the single-processing detector across JPEG quality factors. SCRMQ1, BOSSColor.

[10] N. Dalgaard, C. Mosquera, and F. Pérez-González. On the role of differentiation for resampling detection. In *IEEE International Conference on Image Processing, ICIP 2010*, pages 1753–1756, 2010.

[11] T. Denemark, M. Boroumand, and J. Fridrich. Steganalysis features for content-adaptive JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 11(8):1736–1746, August 2016.

[12] H. Farid. Blind inverse gamma correction. *IEEE Transactions on Image Processing*, 10(10):1428–1433, 2001.

[13] X. Feng, I. J. Cox, and G. Doërr. Normalized energy density-based forensic detection of resampled images. *IEEE Transactions on Multimedia*, 14(3):536–545, 2012.

[14] J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2011.

[15] A. C. Gallagher and T.-H. Chen. Image authentication by detecting traces of demosaicing. In *IEEE Workitorial on Vision of the Unseen (in conjunction with CVPR)*, 2008.

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

[17] V. Holub and J. Fridrich. Phase-aware projection model for steganalysis of JPEG images. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[18] M. Kirchner. Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In *ACM Multimedia and Security Workshop*, pages 11–20, Oxford, UK, 2008.

[19] M. Kirchner. Linear row and column predictors for the analysis of resized images. In *MM&Sec'10, Proceedings of the 2010 ACM SIGMM Multimedia & Security Workshop*, pages 13–18. ACM Press, 2010.

[20] M. Kirchner and J. Fridrich. On detection of median filtering in images. In *Proc. SPIE, Electronic Imaging, Media Forensics and Security XII*, volume 7542, pages 10 1–12, San Jose, CA, January 17–21 2010.

[21] M. Kirchner and T. Gloe. On resampling detection in re-compressed images. In *Information Forensics and Security, 2009. WIFS 2009. First IEEE International Workshop on*, pages 21–25, Dec 2009.

[22] H. Li, W. Luo, X. Qui, and J. Huang. Identification of various image operations using residual-based features. *IEEE Transactions on Information Forensics and Security*, August 2016.

[23] B. Mahdian and S. Saic. Blind authentication using periodic properties of interpolation. *Information Forensics and Security, IEEE Transactions on*, 3(3):529–538, Sept 2008.

[24] M. K. Mihcak, I. Kozintsev, and K. Ramchandran. Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3253–3256, Phoenix, AZ, March 15–19, 1999.

[25] R. Neelamani, R. de Queiroz, Z. Fan, S. Dash, and R. G. Baraniuk. JPEG compression history estimation for color images. *IEEE Transactions on Image Processing*, 15(6):1365–1378, 2006.

[26] H. C. Nguyen and S. Katzenbeisser. Robust resampling detection in digital images. In B. De Decker and D. W. Chadwick, editors, *Communications and Multimedia Security, CMS 2012*, number 7394 in Lecture Notes in Computer Science, pages 3–15, Berlin, Heidelberg, 2012. Springer-Verlag.

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

[28] S. Prasad and K. R. Ramakrishnan. On resampling detection and its application to detect image tampering. In *International Conference on Multimedia and EXPO, ICME 2006*, pages 1325–1328, 2006.

[29] Z. Qu, W. Luo, and J. Huang. A convolutive mixing model for shifted double JPEG compression with application to passive image authentication. In *2008 IEEE International Conference on Acoustics, Speech,*

and Signal Processing, ICASSP 2008, pages 1661–1664, 2008.

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

[31] M. Stamm and K. J. R. Liu. Blind forensics of contrast enhancement in digital images. In *2008 IEEE International Conference on Image Processing, ICIP 2008*, pages 3112–3115, 2008.

[32] M. Stamm and K. J. R. Liu. Forensic estimation and reconstruction of a contrast enhancement mapping. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2010*, pages 1698–1701, 2010.

[33] A. Swaminathan, M. Wu, and K. J. R. Liu. Digital image forensics via intrinsic fingerprints. *IEEE Transactions on Information Forensics and Security*, 3(1):101–117, 2008.

[34] F. Uccheddu, A. de Rosa, A. Piva, and M. Barni. Detection of resampled images: Performance analysis and practical challenges. In *18th European Signal Processing Conference (EUSIPCO 2010)*, pages 1675–1679, 2010.

[35] L. Verdoliva, D. Cozzolino, and G. Poggi. A feature-based approach for image tampering detection and localization. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 149–154, 2014.

[36] R. Wang and X. Ping. Detection of resampling based on singular value decomposition. In *Image and Graphics, 2009. ICIG '09. Fifth International Conference on*, pages 879–884, Sept 2009.

[37] H. Yao, S. Wang, and X. Zhang. Detect piecewise linear contrast enhancement and estimate parameters using spectral analysis of image histogram. In *IET International Communication Conference on Wireless Mobile and Computing, CCWMC 2009*, pages 94–97, 2009.

[38] H.-D. Yuan. Blind forensics of median filtering in digital images. *Information Forensics and Security, IEEE Transactions on*, 6(4):1335–1345, Dec 2011.

Author Biography

Mehdi Boroumand received his B.S. degree in electrical engineering from the K. N. Toosi University of Technology, Iran, in 2004 and his M.S. degree in electrical engineering from the Sahand University of Technology, Iran in 2007. From 2007 to 2013 he worked in the industry at companies like Ericsson, ZTE and MTN. He is currently pursuing the Ph.D. degree in electrical engineering with Binghamton University. His areas of research include steganography, steganalysis, digital image forensics, and machine learning.

Jessica Fridrich is Distinguished Professor of Electrical and Computer Engineering at Binghamton University. She received her PhD in Systems Science from Binghamton University in 1995 and MS in Applied Mathematics from Czech Technical University in Prague in 1987. Her main interests are in steganography, steganalysis, and digital image forensics. Since 1995, she has received 20 research grants totaling over $11 mil that lead to more than 180 papers and 7 US patents.