Exposing Digital Forgeries by Detecting Duplicated Image Regions

Alin C Popescu and Hany Farid†
Department of Computer Science
Dartmouth College
Hanover NH 03755

Abstract

We describe an efficient technique that automatically detects duplicated regions in a digital image. This technique works by first applying a principal component analysis to small fixed-size image blocks to yield a reduced dimension representation. This representation is robust to minor variations in the image due to additive noise or lossy compression. Duplicated regions are then detected by lexicographically sorting all of the image blocks. We show the efficacy of this technique on credible forgeries, and quantify its robustness and sensitivity to additive noise and lossy JPEG compression.

† Corresponding author: H. Farid, 6211 Sudikoff Lab, Computer Science Department, Dartmouth College, Hanover, NH 03755 USA (email: farid@cs.dartmouth.edu; tel/fax: 603.646.2761/603.646.1672). This work was supported by an Alfred P. Sloan Fellowship, a National Science Foundation CAREER Award (IIS-99-83806), a departmental National Science Foundation Infrastructure Grant (EIA-98-02068), and under Award No. 2000-DT-CS-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security (points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security).

1 Introduction

Sophisticated digital cameras and photo-editing software packages are becoming ubiquitous. As a result, it has become relatively easy to manipulate digital images and create forgeries that are difficult to distinguish from authentic photographs. A common manipulation in tampering with an image is to copy and paste portions of the image to conceal a person or object in the scene. If the splicing is imperceptible, little concern is typically given to the fact that identical (or virtually identical) regions are present in the image.

In this paper, we present a technique that can efficiently detect and localize duplicated regions in an image. This technique works by first applying a principal component analysis (PCA) on small fixed-size image blocks to yield a reduced dimension representation. This representation is robust to minor variations in the image due to additive noise or lossy compression. Duplicated regions are then detected by lexicographically sorting all of the image blocks. A similar method for detecting duplicated regions based on lexicographic sorting of DCT block coefficients was proposed in [3]. While both methods employ a similar approach, the data-driven PCA basis may better capture discriminating features. We show the efficacy of this technique on credible forgeries, and quantify its robustness and sensitivity to additive noise and lossy JPEG compression.

2 Detecting Duplicated Regions

Given an image with N pixels our task is to determine if it contains duplicated regions of unknown location and shape. An exhaustive approach that would examine every possible pair of regions would have an exponential complexity in the number of image pixels. Such an approach is obviously computationally prohibitive.

A more efficient algorithm might look for duplication of small fixed-sized blocks [1]. By stringing each such block into a vector and lexicographically sorting all image blocks, identical blocks correspond to adjacent pairs in the sorted list. The primary cost of this algorithm would be the lexicographic sorting, yielding a complexity of $O(N \log N)$, since the number of image blocks is proportional to the number of image pixels, N. Note that this is a significant improvement over the brute-force exponential algorithm. The drawback of this approach, however, is that it is sensitive to small variations between duplicated regions due to, for example, additive noise or lossy compression. We describe next an algorithm that overcomes this limitation while retaining its efficiency.

Consider a grayscale image with N pixels (we discuss below how this algorithm extends to color images). An image is tiled with overlapping blocks of b pixels ($\sqrt{b} \times \sqrt{b}$ pixels in dimension), each of which are assumed to be considerably smaller than the size of the duplicated regions to be detected. Let $\vec{x}_i$ $i = 1, \ldots, N_b$ denote these blocks in vectorized form, where $N_b = (\sqrt{N} - \sqrt{b} + 1)^2$. We now consider an alternate representation of these image blocks based on a principal component analysis (PCA) [1]. Assume that the blocks $\vec{x}_i$ are zero-mean [2], and compute the covariance matrix as:

$$C = \sum_{i=1}^{N_b} \vec{x}_i \vec{x}_i^T. \quad (1)$$

The eigenvectors, $\vec{e}_j$, of the matrix C, with corresponding eigenvalues, λ_j, satisfying:

$$C\vec{e}_j = \lambda_j \vec{e}_j, \quad (2)$$

[1] We assume that the size of the blocks is considerably smaller than the duplicated region to be detected.

[2] If the blocks, $\vec{x}_i$, are not zero-mean, then the mean, $\vec{\mu} = 1/N_b \sum_{i=1}^{N_b} \vec{x}_i$, should be subtracted from each block, $\vec{x}_i - \vec{\mu}$.

define the principal components, where $j = 1, \ldots, b$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_b$. The eigenvectors, $\vec{e}_j$, form a new linear basis for each image block, $\vec{x}_i$:

$$\vec{x}_i \quad = \quad \sum_{j=1}^{b} a_j \vec{e}_j, \tag{3}$$

where $a_j = \vec{x}_i^T \vec{e}_j$, and $\vec{a}_i = (a_1 \ \ldots \ a_b)$ is the new representation for each image block.

The dimensionality of this representation can be reduced by simply truncating the sum in Equation (3) to the first N_t terms. Note that the projection onto the first N_t eigenvectors of the PCA basis gives the best N_t-dimensional approximation in the least squares sense (if the distribution of the $\vec{x}_i$s is a multi-dimensional Gaussian [1]). This reduced dimension representation, therefore, provides a convenient space in which to identify similar blocks in the presence of corrupting noise, as truncation of the basis will remove minor intensity variations.

The detection algorithm proceeds as follows. First, to further reduce minor variations due to corrupting noise, the reduced dimension representation of each image block, $\vec{a}_i$, is component-wise quantized, $\lfloor \vec{a}_i / Q \rfloor$, where the positive integer Q denotes the number of quantization bins [3]. A $N_b \times b$ matrix is constructed whose rows contain these quantized coefficients. Let the matrix S be the result of lexicographically sorting the rows of this matrix in column order. Let $\vec{s}_i$ denote the i^{th} row of this sorted matrix, and let the tuple (x_i, y_i) denote the block's image coordinates (top-left corner) that corresponds to $\vec{s}_i$. Consider next all pairs of rows $\vec{s}_i$ and $\vec{s}_j$, whose row distance, $|i - j|$, in the sorted matrix S is less than a specified threshold. The offset, in the image, of all such pairs is given by:

$$\begin{aligned} (x_i - x_j, y_i - y_j) \quad & \text{if} \quad x_i - x_j > 0 \\ (x_j - x_i, y_i - y_j) \quad & \text{if} \quad x_i - x_j < 0 \\ (0, |y_i - y_j|) \quad & \text{if} \quad x_i = x_j \end{aligned}$$

From a list of all such offsets, duplicated regions in the image are detected by noting the offsets with high occurrence. For example a large duplicated region will consist of many smaller blocks, each of these blocks will appear in close proximity to each other in the lexicographically sorted matrix, and will have the same offset. In order to avoid false hits due to uniform intensity areas, offset magnitudes below a specified threshold are ignored. See Appendix A for a detailed step-by-step algorithm.

The results of this detection can be visualized by constructing a duplication map — a zero image of the same size as the original is created, and all pixels in a region believed to be duplicated are assigned a unique grayscale value. The complexity of this algorithm, dominated by the lexicographic sorting, is $O(N_t N \log N)$, where N_t is the dimension of the PCA reduced representations and N is the total number of image pixels.

There are at least two ways in which this algorithm can be extended to color images. The simplest approach is to independently process each color channel (e.g., RGB) to yield three duplication maps. The second approach is to apply PCA to color blocks of size $3b$, and proceed in the same way as described above.

[3] For simplicity we a use constant number of quantization bins, although it might be more appropriate to use more bins for the coordinates with higher variance, and fewer bins for the lower variance coordinates.

3 Results

Shown in Figures 1-3 are an original and tampered image. The tampering consisted of copying and pasting a region in the image to conceal a person or object. Shown in the lower portion of these figures are the outputs of our detection algorithm as applied to the tampered image saved with JPEG quality factors between 50 and 100. In each map, the two duplicated regions are shown with different grayscale values. In all of these examples, all parameters were set to: $b = 64$, $\epsilon = 0.01$, $Q = 256$, $N_n = 100$, $N_f = 128$, $N_d = 16$ (see Appendix A for details). Truncation of the PCA basis typically reduces the dimension from 64 to 32. The average runtime for one color channel of a 512×512 image running on a 3 GHz processor, is approximately 10 seconds. For visualization purposes, the duplication map was (1) dilated then eroded to eliminate holes in the duplicated regions, and (2) eroded then dilated to eliminate spurious pairs of duplicated blocks. A disk-shaped structuring element with a radius of 20 pixels was employed for these morphologic operations [4].

To quantify the robustness and sensitivity of our algorithm we constructed a database of 100 color images of size 512×512 pixels. These images were cropped from larger 2000×3008 images taken with a Nikon D100 digital camera. In each image, a random square region was copied and pasted onto a random non-overlapping position in the image. Each image was then either JPEG compressed with varying quality factors, or corrupted with additive noise with varying signal to noise ratios (SNR). Shown on the top row of Figure 4, for example, are four images with duplicated regions of size 32×32, 64×64, 96×96, and 128×128 — the first two images were compressed with JPEG quality 85 and 65, and the other two images were corrupted with additive noise with a SNR of 36db and 29db. Shown on the bottom row of Figure 4 are the duplication maps returned when running our algorithm on each image's green channel. In these examples, and those described below, all parameters were set to: $b = 64$, $\epsilon = 0.01$, $Q = 256$, $N_n = 100$, $N_f = 128$, $N_d = 16$. Shown in Figure 5 is the detection accuracy and false positive rate as a function of JPEG compression quality. Note that the accuracy is, in general, very good, except for small block sizes and low JPEG qualities. Note also that the average number of false positives (regions incorrectly labeled as duplicated) is relatively low. Shown in Figure 6 is the detection accuracy and false positive rate as a function of signal to noise ratio (SNR) of additive white Gaussian noise. As in the previous example, the detection rates are nearly perfect, except for small block sizes and low SNR.

4 Discussion

We have presented an efficient and robust technique that automatically detects duplicated regions in an image. This technique works by first applying a principal component analysis (PCA) on small fixed-size image blocks to yield a reduced dimension representation that is robust to minor variations in the image due to additive noise or lossy compression. Duplicated regions are then detected by lexicographically sorting all of the image blocks. We have shown the effectiveness of this technique on plausible forgeries, and have quantified its sensitivity to JPEG lossy compression and additive noise — we find that detection is possible even in the presence of significant amounts of corrupting noise.

We have also been developing other techniques for detecting traces of digital tampering in images [2, 6, 7, 8]. Each technique in this suite works in the complete absence of digital watermarks or signatures offering a complementary approach for image authentication. There is little doubt that counter-measures will be created to foil each of these techniques. Our hope, however, is that our tools, as well as those of others [3, 5], will make it increasingly harder to create credible digital forgeries that will simultaneously foil each of the detection schemes.

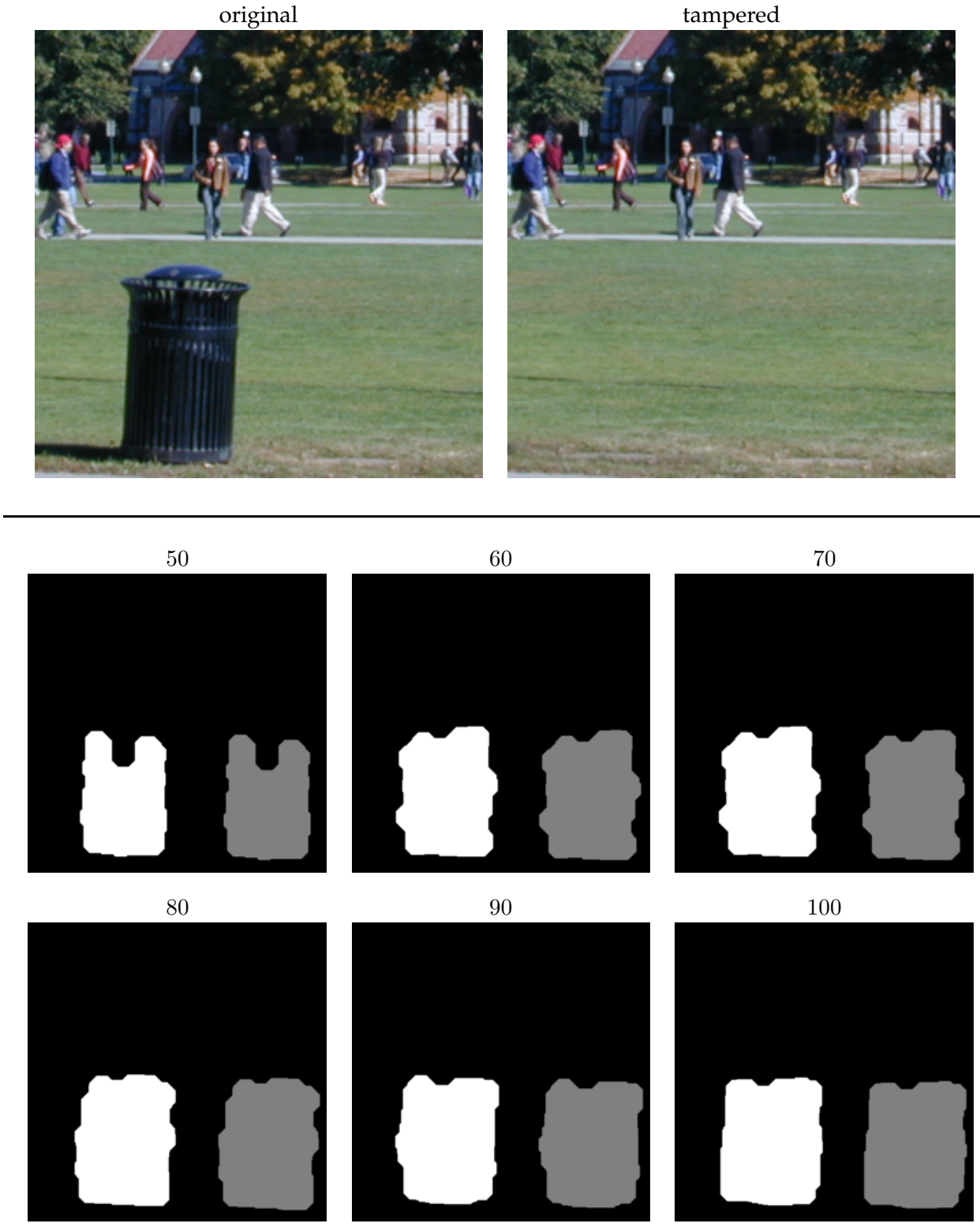

Figure 1: Shown are an original and tampered image. Shown below are the output duplication maps from the green channel of the tampered image saved with JPEG qualities ranging between 50 and 100.

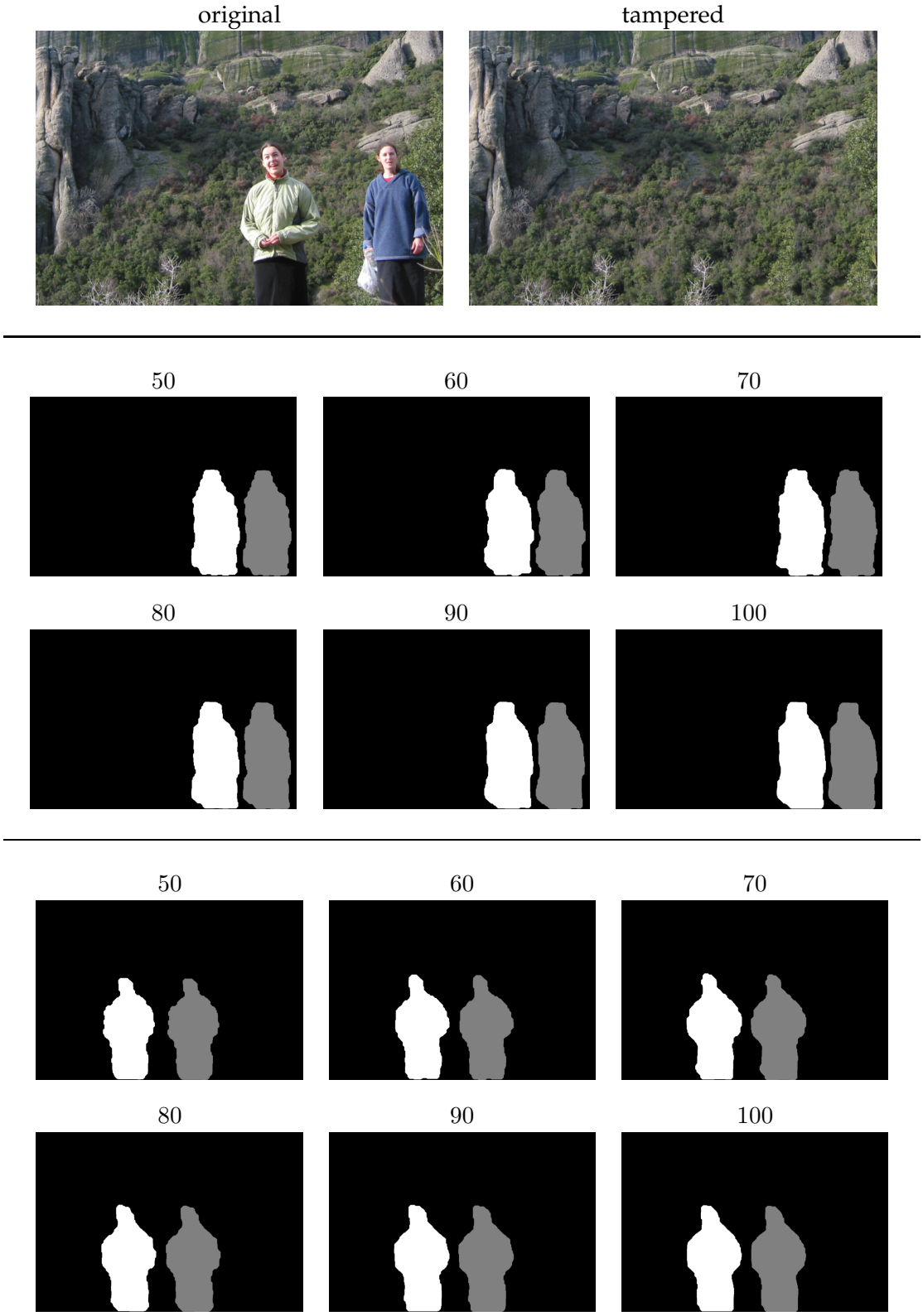

Figure 2: Shown are an original and tampered image. Shown below are the output duplication maps (corresponding to different regions used to conceal each person) from the green channel of the tampered image saved with JPEG qualities ranging between 50 and 100.

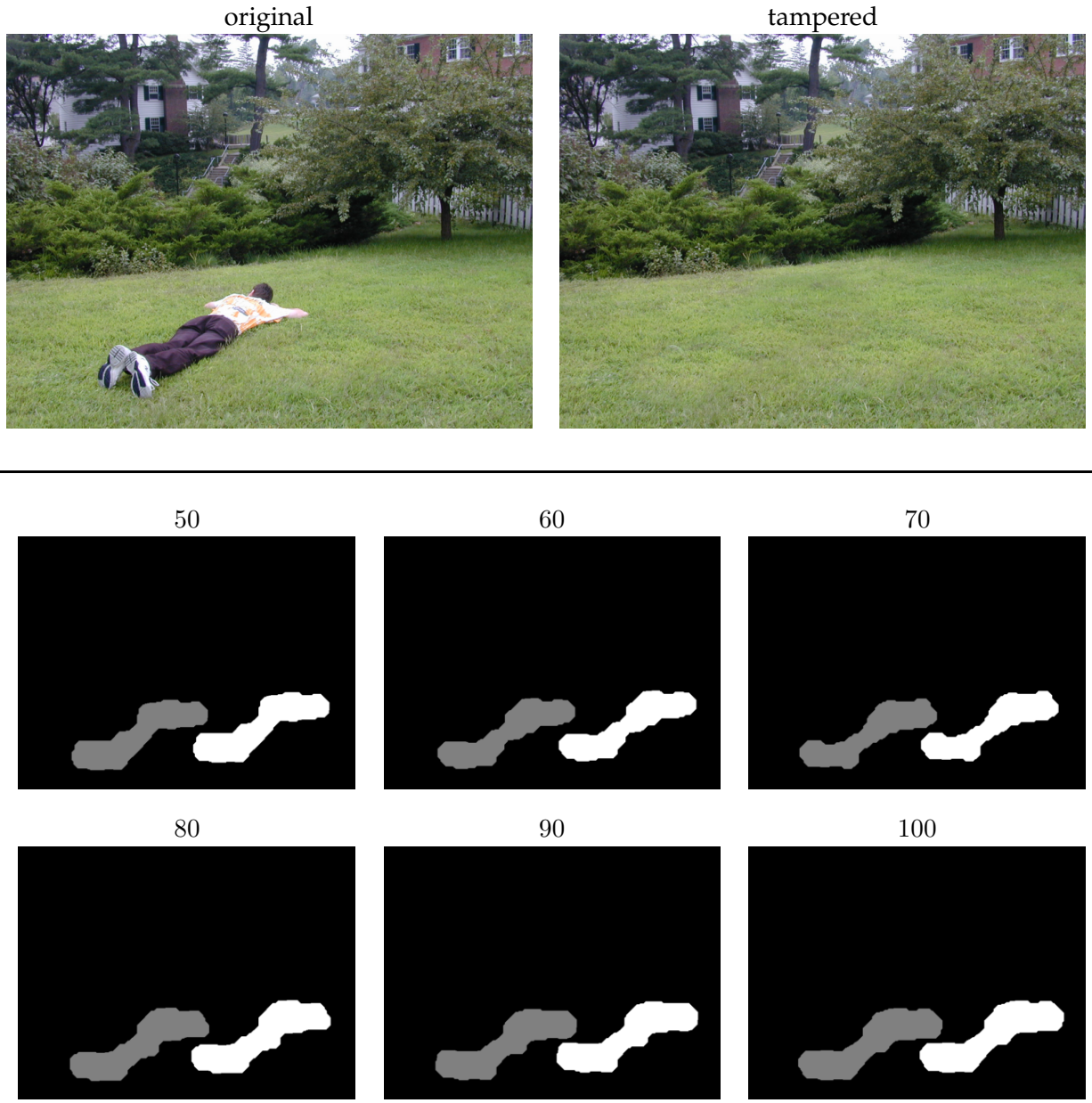

Figure 3: Shown are an original and tampered image. Shown below are the output duplication maps from the green channel of the tampered image saved with JPEG qualities ranging between 50 and 100.

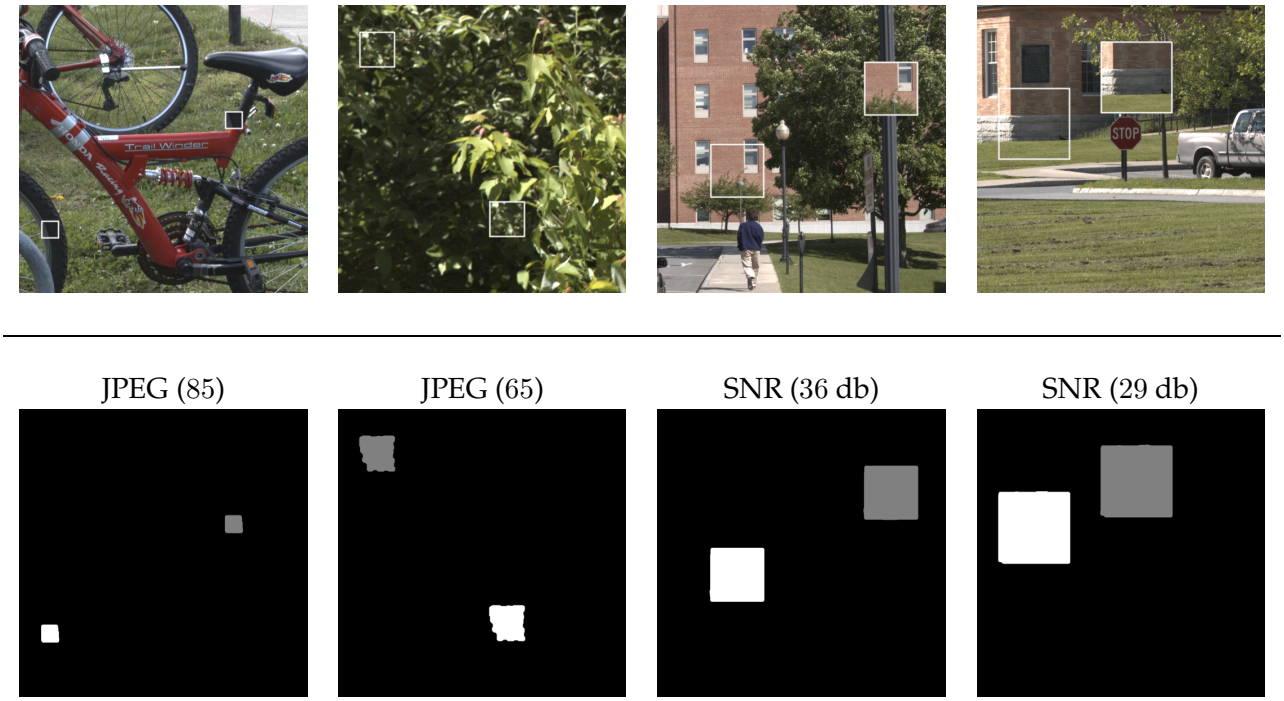

Figure 4: Shown on the top row are four images with duplicated regions of size 32×32, 64×64, 96×96, and 128×128, after having been compressed (first and second images) or corrupted with additive noise (third and fourth images). Shown below are the duplication maps returned when running our algorithm on each image's green channel.

References

[1] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[2] H. Farid and S. Lyu. Higher-order wavelet statistics and their application to digital forensics. In *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, Wisconsin, 2003.

[3] J. Fridrich, D. Soukal, and J. Lukáš. Detection of copy-move forgery in digital images. In *Proceedings of Digital Forensic Research Workshop*, August 2003.

[4] R.C. Gonzalez and R.E. Wood. *Digital Image Processing*. Addison Wesley Publishing Company, 1992.

[5] J. Lukas and J. Fridrich. Estimation of primary quantization matrix in double compressed JPEG images. In *Digital Forensic Research Workshop*, Cleveland, Ohio, August 2003.

[6] A. C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of re-sampling. *IEEE Transactions on Signal Processing*, 2004. (to appear).

[7] A. C. Popescu and H. Farid. Statistical tools for digital forensics. In *Proceedings of the* 6^{th} *Information Hiding Workshop*, May 2004.

[8] A. C. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 2005. (in review).

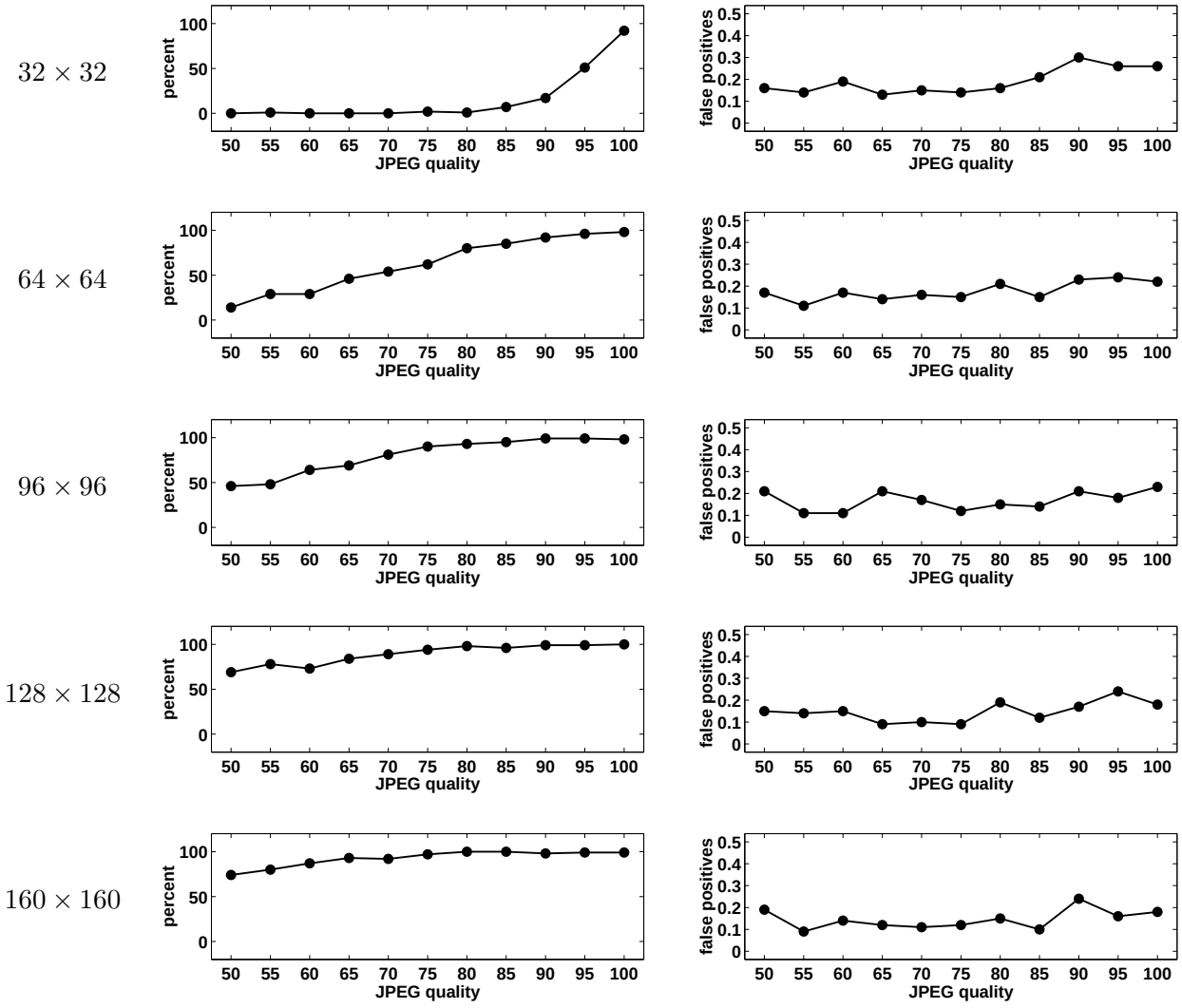

Figure 5: Shown in the left column are the average detection accuracies as a function of the JPEG compression quality. Shown in the right column are the average number of false positives as a function of the JPEG compression quality. Each row corresponds to duplicated blocks of size ranging from 32×32 to 160×160 pixels. Each data point corresponds to an average over 100 images.

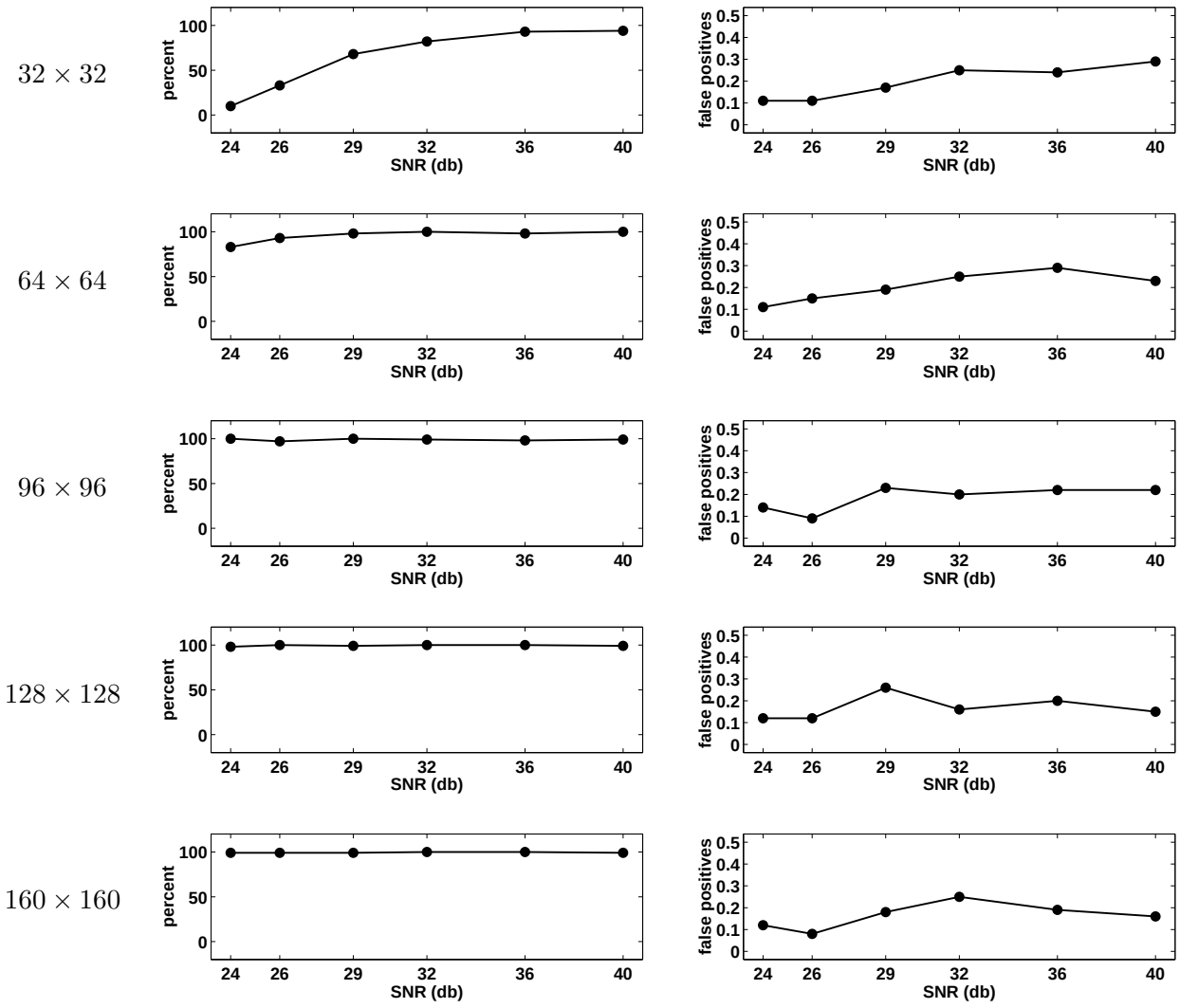

Figure 6: Shown in the left column are the average detection accuracies as a function of the SNR of added white Gaussian noise. Shown in the right column are the average number of false positives as a function of the SNR. Each row corresponds to duplicated blocks of sizes ranging from 32×32 to 160×160 pixels. Each data point corresponds to an average over 100 images.

A Duplication Detection Algorithm

1. Let N be the total number of pixels in a grayscale or color image
2. Initialize the parameters:
   - b: number of pixels per block ($\sqrt{b} \times \sqrt{b}$ pixels in dimension) – there are $N_b = (\sqrt{N} - \sqrt{b} + 1)^2$ such blocks
   - ϵ: fraction of the ignored variance along the principal axes
   - Q: number of quantization bins
   - N_n: number of neighboring rows to search in the lexicographically sorted matrix
   - N_f: minimum frequency threshold
   - N_d: minimum offset threshold
3. Using PCA, compute the new N_t-dimensional representation, $\vec{a}_i$, $i = 1, \ldots, N_b$, of each b pixel image block (for color images: (1) analyze each color channel separately; or (2) build a single color block of size $3b$ pixels). The value of N_t is chosen to satisfy: $1 - \epsilon = \frac{\sum_{i=1}^{N_t} \lambda_i}{\sum_{i=1}^{b} \lambda_i}$, where λ_i are the eigenvalues as computed by the PCA.
4. Build a $N_b \times b$ matrix whose rows are given by the component-wise quantized coordinates: $\lfloor \vec{a}_i / Q \rfloor$.
5. Sort the rows of the above matrix in lexicographic order to yield a matrix S. Let $\vec{s}_i$ denote the rows of S, and let (x_i, y_i) denote the position of the block's image coordinates (top-left corner) that corresponds to $\vec{s}_i$.
6. For every pair of rows $\vec{s}_i$ and $\vec{s}_j$ from S such that $|i - j| < N_n$, place the pair of coordinates (x_i, y_i) and (x_j, y_j) onto a list.
7. For all elements in this list, compute their offsets, defined as:

$$\begin{aligned} (x_i - x_j, y_i - y_j) \quad & \text{if} \quad x_i - x_j > 0 \\ (x_j - x_i, y_i - y_j) \quad & \text{if} \quad x_i - x_j < 0 \\ (0, |y_i - y_j|) \quad & \text{if} \quad x_i = x_j \end{aligned}$$

8. Discard all pairs of coordinates with an offset frequency less than N_f.
9. Discard all pairs whose offset magnitude, $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, is less than N_d.
10. From the remaining pairs of blocks build a duplication map by constructing a zero image of the same size as the original, and coloring all pixels in a duplicated region with a unique grayscale intensity value.