NOISE FEATURES FOR IMAGE TAMPERING DETECTION AND STEGANALYSIS

Hongmei Gou, Ashwin Swaminathan and Min Wu

ECE Department, University of Maryland, College Park, USA

ABSTRACT

With increasing availability of low-cost image editing softwares, the authenticity of digital images can no longer be taken for granted. Digital images have also been used as cover data for transmitting secret information in the field of steganography. In this paper, we introduce a new set of features for multimedia forensics to determine if a digital image is an authentic camera output or if it has been tampered or embedded with hidden data. We perform such image forensic analysis employing three sets of statistical noise features, including those from denoising operations, wavelet analysis, and neighborhood prediction. Our experimental results demonstrate that the proposed method can effectively distinguish digital images from their tampered or stego versions.

Index Terms— Multimedia forensics, Tampering detection, steganalysis, noise features.

1. INTRODUCTION

In the modern information era, digital images have been widely used in a growing number of applications related to military, intelligence, surveillance, law enforcement, and commercial applications. Meanwhile, with the growing number of low-cost easy-to-use image editing softwares, the authenticity of an image can no longer be taken for granted. In the field of steganography, digital images have also been used as cover data for transmitting secret information, and a number of data hiding algorithms have been developed for such stego purposes. Distinguishing digital images as direct camera outputs from their tampered or stego versions involves establishing the integrity of digital images. Although semi-fragile watermarking [1] and robust hashing have been proposed as solutions to image integrity establishment, they would require the watermark to be inserted or the hash to be generated at the time of image creation; but most digital cameras in the market still lack such capability. Hence, there is a strong need as part of the emerging field of multimedia forensics to develop *non-intrusive* methodologies for tampering detection and steganalysis.

Existing methods for image tampering detection and steganalysis can be classified into two categories. In the first category, manipulation-specific methods are developed with the aim of detecting a particular type of tampering operation such as compression, filtering, gamma correction, and re-sampling [2], or for identifying the presence of hidden data embedded using a specific steganographic embedding algorithm [3, 4]. Although these methods work well in detecting a particular type of tampering or steganographic embedding operation, it would require an exhaustive search over all the possible kinds of operations to establish the integrity of a digital image. In the second category, classifier-based approaches are proposed for generic tampering detection [5] and for blind steganlysis [6, 7] on digital images. These techniques provide a framework for universal image forensic analysis independent of the nature of tampering or stego manipulations. Features such as image quality metrics [5] and higher-order wavelet statistics [7] are utilized to build the classifiers.

In this work, we propose using statistical noise features of digital images to discriminate direct camera outputs from their tampered and stego versions. The basic idea behind our approach is that image manipulations, such as tampering and steganographic embedding, change the image noise statistics in specific ways, and such changes can be utilized to perform forensic analysis. Specifically, we characterize image noise from multiple perspectives via image de-noising, wavelet analysis, and neighborhood prediction; and obtain statistical features from each noise characterization. As we will show later in this paper, a classifier built using the proposed noise features can effectively distinguish digital images from their tampered or stego versions.

The rest of the paper is organized as follows. The details of the proposed noise features are described in Section 2. In Section 3, we present experimental results on applying the proposed noise features to image tampering detection and steganalysis. The final conclusions are drawn in Section 4.

2. STATISTICAL NOISE FEATURE EXTRACTION

In this section, we discuss methodologies to extract image noise features for tampering detection and steganalysis. We characterize image noise from the following three aspects [8]. For the first set of features, we apply denoising algorithms to an image to obtain estimates of image noise. We extract the second set of features based on Gaussian fitting errors of wavelet coefficients. Finally, we characterize image noise through neighborhood prediction and use the prediction error for extracting the third set of features.

Noise Features from Denoising Algorithms: To extract image noise features, we first utilize image denoising algorithms. As shown in Fig. 1(a), given an image I, denoising operation is applied to obtain its denoised version I_D. The estimated image noise n_I at the pixel location (i, j) is then

Email contact: {hmgou, ashwins, minwu}@eng.umd.edu.

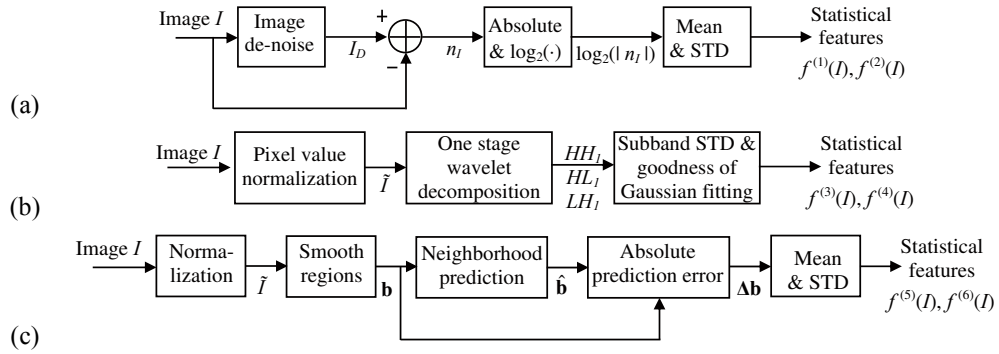

Fig. 1. Statistical noise feature extraction via (a) image denoising, (b) wavelet analysis, and (c) neighborhood prediction.

found by pixel-wise subtraction $n_I(i,j) = I(i,j) - I_D(i,j)$. Let $e(i,j) = \log_2(|n_I(i,j)|)$. The mean and the standard deviation of $\{e(i,j)\}$ form the first set of features:

$$\begin{cases} f^{(1)}(I) = \mu_e = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} e(i,j), \\ f^{(2)}(I) = \sigma_e = \left(\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(e(i,j) - f^{(1)}(I)\right)^2\right)^{\frac{1}{2}} \end{cases} \quad (1)$$

where M and N indicate the size of the image I.

To capture the different aspects of noise, we apply four different denoising algorithms to an image: linear filtering with an averaging filter, linear filtering with a Gaussian filter, median filtering, and Wiener adaptive image denoising. Low-pass linear filtering using averaging or Gaussian filters helps model the high-frequency noise, non-linear median filtering addresses the "salt and pepper" noise, and adaptive methods such as Wiener filtering can tailor noise removal to the local pixel variance. In our experiments, we use an averaging filter of size 3×3, a Gaussian low-pass filter of the same size and with a standard deviation $\sigma = 0.5$, a median filter of size 3×3, and adaptive Wiener denoising with two neighborhood sizes 3×3 and 5×5, respectively. Using these denoising settings, we obtain five denoised versions for image I. For each of them, we extract the two features in (1) from each of the three color components (RGB), and therefore arrive at a total of $5 \times 2 \times 3 = 30$ features.

Noise Features from Wavelet Analysis: We obtain the second set of noise features via wavelet analysis. After one stage 2-D wavelet decomposition, an input image is decomposed to four subbands, namely, low-low (LL), low-high (LH), high-low (HL), and high-high (HH) subbands. Among these four subbands, the LL subband contains low-frequency components, while the other three are for high-frequency components. In literature, it has been observed that for a large class of images, the wavelet coefficients in the LH, HL, and HH subbands do not follow a Gaussian distribution [9]. This is because the spatial structure of these images consists of smooth areas interspersed with occasional edges, and therefore coefficients in the high-frequency subbands are sharply peaked at zero with broad tails. When applying tampering operations or data hiding to an image, such non-Gaussian property of the high-frequency wavelet coefficients may be affected. The noise strength may also be changed due to the tampering operations or the data hiding.

Based on above analysis, we extract statistical noise features in the wavelet domain as follows, and the basic modules are shown in Fig. 1(b). Given an image I, we first normalize it to be $\tilde{I}$ with unit energy, i.e.,

$$\tilde{I}(i,j) = \frac{I(i,j)}{\left(\frac{1}{MN}\sum_{k=1}^{M}\sum_{l=1}^{N} I(k,l)^2\right)^{\frac{1}{2}}}. \quad (2)$$

Then, we perform one stage 2-D wavelet decomposition to $\tilde{I}$ and obtain its three high-frequency subbands HH, HL, LH. After that, for each of these three subbands, we calculate the mean μ_Y and the standard deviation σ_Y of the wavelet subband coefficients $\{Y(u,v)\}$. We take the standard deviation as our third statistical noise feature:

$$f^{(3)}(I) = \sigma_Y. \quad (3)$$

With the computed sample mean μ_Y and variance σ_Y^2, we arrive at a Gaussian distribution $\mathbf{N}(\mu_Y, \sigma_Y^2)$, and further quantify the goodness of fitting this Gaussian distribution to the distribution of $\{Y(u,v)\}$. Let $p(y)$ and $q(y)$ denote the probability density functions (PDFs) of the Gaussian distribution $\mathbf{N}(\mu_Y, \sigma_Y^2)$ and the distribution of the subband wavelet coefficients $\{Y(u,v)\}$, respectively. We quantify the goodness of Gaussian fitting by measuring the distance between $p(y)$ and $q(y)$ as $\delta_1 = \int |p(y) - q(y)|dy$, whose discrete-summation approximation is taken as the fourth statistical noise feature:

$$f^{(4)}(I) = \sum_i |p(y_i) - q(y_i)|\Delta y, \quad (4)$$

where i is the index of histogram bins, Δy is the length of each bin, $p(y_i)$ is the value of the Gaussian PDF $p(y)$ at the center of the i^{th} bin, and $q(y_i)$ is the count of the i^{th} bin normalized toward a valid PDF ($\sum_i q(y_i)\Delta y = 1$). The two features $f^{(3)}(I)$ and $f^{(4)}(I)$ are extracted from each of the three high-frequency subbands and for each of the three color components, and therefore a total of $2 \times 3 \times 3 = 18$ features are obtained in the wavelet domain.

Noise Features from Neighborhood Prediction: Most images consist of some smooth regions, where pixel values can be predicted from certain neighboring pixels with high accuracy. However, when smooth regions of an image are contaminated by noise, non-trivial prediction errors may be resulted in during the neighborhood prediction. Therefore, we characterize image noise in terms of the neighborhood prediction error in the smooth regions, and then extract the third set of noise features from it.

In Fig. 1(c), we show the basic modules of extracting statistical noise features via neighborhood prediction. Given an image I, we first identify its smooth region according to local image gradient values. Before calculating the gradient values, we still normalize I to be $\tilde{I}$ with unit energy as in (2). Comparing horizontal/vertical image gradient values g_h and g_v with a threshold t_g, we identify pixels in the smooth region as those of both a small horizontal gradient and a small vertical gradient. Setting a threshold t_i on the pixel intensity value, the smooth region is further partitioned to be a dark smooth region and a bright smooth region. In our test, we set the gradient threshold $t_g = 0.2$ and the intensity threshold t_i as the median of the pixel intensity values in $\tilde{I}$.

For each of the two smooth regions, we now perform neighborhood prediction. We predict each pixel value b_i in a given region using a linear model on its eight-connected neighborhood $\{a_{i,1} - a_{i,8}\}$: $\hat{b}_i = \sum_{k=1}^{8} x_k a_{i,k}$. Here, $x_k \geq 0$ is the weight associated with $a_{i,k}$, and the non-negative constraint indicates positive correlation between b_i and its neighbors. Given a region with N pixels, we denote its N pixel values as a column vector $\mathbf{b} = [b_1, b_2, \ldots, b_N]^T$, and the non-negative weight coefficients as a column vector $\mathbf{x} = [x_1, x_2, \ldots, x_8]^T$. Further, we represent the eight neighbors of each pixel as a row vector, and organize all of them as a matrix A of size $N \times 8$. The estimation of the weight coefficients $\mathbf{x}$ can then be formulated as a non-negative least-squares problem, $\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|^2$ subject to $x_k \geq 0, k = 1, 2, \ldots, 8$, and solved using the algorithm in [10]. After that, we calculate the absolute prediction errors $\Delta\mathbf{b} = |\hat{\mathbf{b}} - \mathbf{b}|$ between the predicted pixel values $\hat{\mathbf{b}} = A\mathbf{x}$ and their original counterparts $\mathbf{b}$. Finally, we take the mean and the standard deviation of $\Delta\mathbf{b}$ as our last two statistical noise features:

$$f^{(5)}(I) = \mu_{\Delta\mathbf{b}}, \quad f^{(6)}(I) = \sigma_{\Delta\mathbf{b}}. \tag{5}$$

For each of the two smooth regions, we extract the two features in (5) for each of the three color components, and therefore a total of $2 \times 2 \times 3 = 12$ features are obtained from the neighborhood prediction.

3. SIMULATION RESULTS AND DISCUSSIONS

In this section, we present experimental results on applying the proposed noise features for image tampering detection and steganalysis. We use 500 images from five different cameras, Canon Powershot A75, FujiFilm Finepix S3000, Minolta DiMage S304, Epson PhotoPC 650, and Nikon E4300, as authentic digital images. These images are captured under completely random conditions with different scenarios and different lighting conditions.

Results for Image Tampering Detection: In the test of tampering detection, the 500 authentic images are first processed to generate 28 different tampered versions per image by (1) average filtering with filter orders {3,5,7}, (2) median filtering with filter orders {3,5,7}, (3) rotating with degrees {1,5,10,20}, (4) re-sampling with percentage {50, 70, 85, 115, 130, 150%}, (5) adding noise of Peak Signal to Noise Ratio (PSNR) {5, 10} dB, (6) gamma correction with $\gamma = \{0.5, 0.7, 0.85, 1.15, 1.3, 1.5\}$, and (7) image sharpening with filter orders {2,4,6,8}. For each of the seven types of tampering operations listed above, we calculate the $30 + 18 + 12 = 60$ noise features discussed in Section 2 for the tampered images. The 60 noise features are also calculated for the 500 authentic images. A ν-support vector machine (SVM) with a radial basis function (RBF) kernel [11] is then used for classifying the authentic images and the tampered images. We randomly choose 250 authentic images along with their corresponding tampered versions for training, and test on the remaining images. Computing the fraction of correctly classified tampered images P_D, and the percentage of authentic images wrongly classified as tampered images P_F, we obtain the receiver operating characteristics (ROC). In Fig. 2(a), we show the ROC averaged over 100 iterations for each of the seven types of tampering, each time with a different selection of the 250 training images. We observe from the figure that the performance is good for most manipulations and the P_D is close to 90% even under a very low P_F close to 5%. This suggests that the proposed noise features can reflect the changes between a direct camera output and its further tampered versions, and effectively detect tampering.

Results for Steganalysis: In the steganalysis test, we use the same 500 authentic images as the cover images, from each of which a number of stego images are generated by embedding random messages of different sizes. In a general scenario, the maximum embedding payload depends on the nature of the cover data and the steganographic embedding method. In our test, we first find the average of the maximum embedding payload across the 500 cover images and then embed messages at 100%, 75%, 50%, and 25% of this value. In our current studies, we consider two popular types of least significant bit (LSB) based steganographic algorithms, namely, F5 [12], and hide4pgp [13].

In the case of F5, the maximum embedding payload averaged over our 500 cover images is around 12 KB. Corresponding to the percentages of 100%, 75%, 50%, and 25%, messages of sizes 12 KB, 9 KB, 6 KB, and 3 KB are embedded into each of the 500 cover images to generate $4 \times 500 = 2000$ stego images in total. For each of the four message sizes, we calculate the 60 noise features for all the 500 stego images in each category. Using these features as well as those

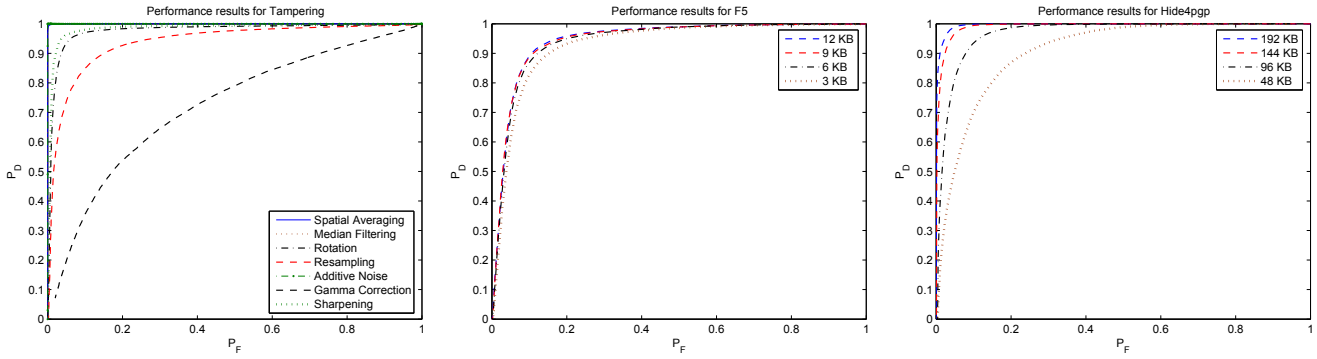

Fig. 2. Receiver Operating Characteristics for (a) different types of tampering operations, and steganographic embedding algorithms (b) F5 and (c) hide4pgp; from left to right.

from the 500 cover images to perform SVM training and testing as in the tampering detection test, we obtain ROC curves as shown in Fig. 2(b). We can see that the performance in discriminating the cover and stego images is good and a P_D close to 90% is obtained for $P_F \approx 10\%$. Further, we notice that the discrimination performance is relatively independent of the embedding rate, suggesting that the proposed noise features can perform accurate steganalysis on F5 even under low embedding payloads. This is because F5 always decreases the magnitude of DCT coefficients when generating stego images. Similarly, we show the ROC curves in Fig. 2(c) for the hide4pgp algorithm under 100%, 75%, 50%, and 25% of the maximum embedding payload averaged over the 500 cover images. In this case, we observe that as the embedded message size increases, the steganalysis performance improves. Under the average maximum embedding payload, we notice that the P_D is close to 98% even for $P_F \approx 10\%$ for most cases, demonstrating the goodness of the proposed noise features for image steganalysis.

4. CONCLUSIONS

In this paper, we have introduced a novel approach for tampering detection and steganalysis on digital images, using three sets of statistical noise features. We apply image denoising algorithms to obtain estimates of image noise and then extract the first set of features from them. Observing that image manipulations affect the non-Gaussian property of wavelet subband coefficients, we extract the second set of features via wavelet analysis. We also perform neighborhood prediction and utilize the prediction error to derive the third set of noise features. Using these three sets of features, we build a robust classifier that can effectively distinguish direct camera outputs from their tampered or stego versions. We have presented detailed simulation results with seven types of tampering operations and with two steganographic embedding algorithms. The obtained results demonstrate the effectiveness of the proposed noise features for image forensic analysis. We believe that the proposed technique would provide a systematic and effective way to establish the integrity of digital images.

5. REFERENCES

[1] J. Fridrich, "Image Watermarking for Tamper Detection," *Proc. of the IEEE ICIP*, vol. 2, pp. 404–408, Oct 1998.

[2] A. C. Popescu and H. Farid, "Statistical Tools for Digital Forensics," *Proc. of Intl. Workshop on Info. Hiding*, Toronto, Canada, & *Lect. Notes in Comp. Sc.*, vol. 3200, pp. 128–147, May 2004.

[3] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG Images: Breaking the F5 Algorithm," *Proc. of Intl. Workshop on Info. Hiding*, 2002.

[4] A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," *Proc. of Intl. Workshop on Info. Hiding*, and *Lecture Notes in Computer Science*, pp. 61–76, 1999.

[5] I. Avcibas, S. Bayram, N. Memon, M. Ramkumar, and B. Sankur, "A Classifier Design for Detecting Image Manipulations," *Proc. of the IEEE ICIP*, vol. 4, pp. 24–27, Oct 2004.

[6] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis Using Image Quality Metrics," *IEEE Trans. on Image Processing*, vol. 12, no. 2, pp. 221–229, Feb 2003.

[7] S. Lyu and H. Farid, "Steganalysis Using Higher-Order Image Statistics," *IEEE Trans. on Info. Forensics and Security*, vol. 1, no. 1, pp. 111–119, Mar 2006.

[8] H. Gou, A. Swaminathan, and M. Wu, "Robust Scanner Identification based on Noise Features," *IS&T SPIE Conf. on Security, Stego., and Watermarking of Multimedia Contents IX*, Jan 2007.

[9] S. G. Chang, B. Yu, and M. Vetterli, "Spatially Adaptive Wavelet Thresholding with Context Modeling for Image Denoising," *IEEE Trans. on Image Processing*, vol. 9, no. 9, pp. 1522–1531, 2000.

[10] C. L. Lawson and R. J. Hanson, *Solving Least-Squares Problems,* Prentice-Hall, 1974.

[11] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–168, 1998.

[12] A. Westfeld, "F5–A Steganographic Algorithm: High Capacity Despite Better Steganalysis," *Proc. of Intl. Workshop on Info. Hiding*, Pittsburgh, PA, April 2001.

[13] *Hide4pgp*, Steganography software available online at `www.heinz-repp.onlinehome.de/Hide4PGP.htm`.