Improved Approaches with Calibrated Neighboring Joint Density to Steganalysis and Seam-Carved Forgery Detection in JPEG Images

QINGZHONG LIU, Sam Houston State University
ZHONGXUE CHEN, Indiana University Bloomington

Steganalysis and forgery detection in image forensics are generally investigated separately. We have designed a method targeting the detection of both steganography and seam-carved forgery in JPEG images. We analyze the neighboring joint density of the DCT coefficients and reveal the difference between the untouched image and the modified version. In realistic detection, the untouched image and the modified version may not be obtained at the same time, and different JPEG images may have different neighboring joint density features. By exploring the self-calibration under different shift recompressions, we propose calibrated neighboring joint density-based approaches with a simple feature set to distinguish steganograms and tampered images from untouched ones. Our study shows that this approach has multiple promising applications in image forensics. Compared to the state-of-the-art steganalysis detectors, our approach delivers better or comparable detection performances with a much smaller feature set while detecting several JPEG-based steganographic systems including DCT-embedding-based adaptive steganography and Yet Another Steganographic Scheme (YASS). Our approach is also effective in detecting seam-carved forgery in JPEG images. By integrating calibrated neighboring density with spatial domain rich models that were originally designed for steganalysis, the hybrid approach obtains the best detection accuracy to discriminate seam-carved forgery from an untouched image. Our study also offers a promising manner to explore steganalysis and forgery detection together.

Categories and Subject Descriptors: I 4.9 [**Image Processing and Computer Vision**]: Applications; K.6.m [**Miscellaneous**]: Insurance and Security

General Terms: Algorithms and Security

Additional Key Words and Phrases: Steganography, steganalysis, YASS, seam carving, calibration, JPEG, neighboring joint density, image tampering

ACM Reference Format:
Qingzhong Liu and Zhongxue Chen. 2014. Improved approaches with calibrated neighboring joint density to steganalysis and seam-carved forgery detection in JPEG images. ACM Trans. Intell. Syst. Technol. 5, 4, Article 63 (December 2014), 30 pages.
DOI: http://dx.doi.org/10.1145/2560365

1. INTRODUCTION

Multimedia forensics is a multiple-disciplinary research field with important impacts on the protection of public safety and enhancement of national security. In multimedia forensics, steganography detection, or steganalysis, and forgery detection are two active areas and are generally separately studied, although both continue to face challenges.

This work is supported by the National Science Foundation, under grant CCF-1318688, and was supported by the National Institute of Justice, U.S. Department of Justice, under the Award No. 2010-DN-BX-K223.
Authors' addresses: Q. Liu, Department of Computer Science, Sam Houston State University; contact author, email: liu@shsu.edu; Z. Chen, Department of Epidemiology and Biostatistics, Indiana University Bloomington; email: zc3@indiana.edu.

DOI: http://dx.doi.org/10.1145/2560365

Steganography, Greek for covered writing, is the art and science of carrying messages in covert channels, aiming to enable secretive communication by embedding data into digital files without attention to the existence of the hidden message. The potential of exploiting steganography for covert dissemination is of increasing concern; a recent espionage case revealed that steganography had been employed by a foreign government intelligence agency [Justice.gov 2010a and 2010b]. Fake photos have been employed for decades, and with various image processing tools, digital images can now be easily forged. Since JPEG has been widely used as a popular image compression standard, there is a heightened urgency to develop effective countermeasures for steganography and forgery in JPEG images.

Quite a few steganographic algorithms/systems have been proposed, including LSB embedding, LSB matching [Mielikainen 2006], spread spectrum steganography [Marvel et al. 1999], Outguess [Provos 2001], F5 [Westfeld 2001], model-based steganography [Sallee 2003, 2005], Steghide [Hetzl and Mutzel 2005], BCH syndrome code-based less detectable JPEG steganography [Sachnev et al. 2009], and highly undetectable steganography (HUGO) [Pevny et al. 2010]. Although these steganographic systems have been successfully steganalyzed [Bayram et al. 2007; Chen and Shi 2008; Fridrich 2004; Fridrich et al. 2011a, 2011b; Fridrich and Kodovsky 2011, 2012; Fu et al. 2006; Gul and Kurugollu 2011; Ker 2004; Kharrazi et al. 2006; Kodovsky et al. 2010; Liu et al. 2005, 2006a, 2006b, 2008a, 2008b, 2008c, 2009a, 2009b, 2010, 2011a, 2011b; Lyu and Farid 2005; Pevny and Fridrich 2007, 2008; Qiao et al. 2013; Shi et al. 2006], the advances in steganography have posed new challenges to steganalyzers [Filler and Fridrich 2010; Filler et al. 2011]. Filler and Fridrich [2011] recently proposed a practical framework of adaptive steganographic systems that optimize the parameters of additive distortion functions and minimize the distortion for ± 1 embedding in the DCT domain. This has greatly improved the art of hiding data in widespread JPEG images. Yet Another Steganographic Scheme (YASS) was designed to be a secure JPEG steganographic algorithm with randomized embedding [Solanki et al. 2007]. By exploring the weakness of YASS steganographic system, Li et al. [2009] presented a simple and efficient detection method by comparing the frequency of zero coefficients of the embedding host blocks and the neighboring blocks in the DCT domain. This detection performance is very promising when the parameter of the big block (B-block) size is small (e.g., the size is set to 9 and 10). However, the detection performance apparently deteriorates if the parameter of the B-block size increases [Li et al. 2009]. Kodovsky et al. [2010] designed 1,234 features to detect YASS and tested 12 different configurations of YASS with a parameter of B-block size no larger than 11. In other words, the detection performance on the YASS steganograms produced by a large parameter of B-block at 12, 13, 14, and 15 was missing [Kodovsky et al. 2010].

Regarding JPEG-based image forgery and its detection, the relevant manipulations, including double JPEG compression, image rescaling, copy-paste, inpainting, and compositing, have been successfully detected [Change et al. 2013; Chen and Hsu 2011; Liu et al. 2011b, 2013; Liu 2011b; Pan and Lyu 2010; Pan et al. 2012; Pevny and Fridrich 2008]. While most image forensics methods target traditional image tampering, seam-carving-based image tampering in JPEG format has been ignored to some extent. Seam carving, an algorithm for image resizing, is known as content-aware scaling, liquid resizing, or liquid rescaling and was designed by Shai Avidan of Mitsubishi Electric Research Labs (MERL) and Ariel Shamir of the Interdisciplinary Center and MERL. It establishes the paths of least importance in an image, called seams; automatically removes them and reduces the image size; or inserts seams to extend the image size [Avidan and Shamir 2007]. Seam carving allows the removal of selected whole objects from photographs. The seam-carving method for content-aware resizing and object removal has been implemented in Adobe Photoshop CS4 [Photoshopsupport.com], GIMP

[liquidrescale.wikidot.com], digiKam [digikam.org], ImageMagick [imageMagick.com], and stand-alone programs such as iResizer [Iresizer.com]. The proliferation of seam-carved images presents a challenge to authorities who require image authentication. Sarkar et al. [2009] employed 324-dimensional Markov features, which were originally developed to detect JPEG-based steganograms by Shi et al. [2006], to distinguish between seam-carved, seam-inserted, and normal images. Fillion and Sharma [2010] designed a method that included benign image reduction, benign image enlargement, and deliberate image reduction to detect seam-carved images and tested their method over a set of images consisting of 1,484 uncompressed images. Unfortunately, the JPEG images were not tested after content-aware manipulation. The detection of seam-carving-based forgery in JPEG images needs extensive further study.

Facing these challenges, we aim to design a method targeting the detection of steganography and seam-carved forgery together in JPEG images rather than separately studying steganalysis and forgery detection. We analyze the neighboring joint density of the DCT coefficients, reveal the difference between untouched and the modified image as well as the calibrations under different shift recompressions, and then propose calibrated neighboring joint-density-based approaches with a simple feature set to distinguish steganograms and tampered images from untouched ones. Our study shows that our approaches have multiple promising applications in image forensics, and compared to the present state of the art, our approaches deliver better or comparable detection performances with a much smaller feature set to detect several steganographic systems, including DCT-embedding-based adaptive steganography and YASS, and to detect seam-carved forgery in JPEG images.

Section 2 briefly describes relevant studies, including DCT-embedding-based adaptive steganography [Filler and Fridrich 2011], YASS [Solanki et al. 2007], a popular detection algorithm specifically designed to break YASS [Li et al. 2009], and our previous study of steganalysis [Liu et al. 2011a]. Section 3 presents our improved approach with calibrated neighboring joint density, and Section 4 describes our improved approach to steganalysis of YASS. Section 5 analyzes the impact of seam carving in JPEG images on modification in the DCT domain and spatial domain and proposes a hybrid approach to the detection by integrating calibrated neighboring joint density and a rich-model-based feature set that was originally designed for steganalysis. Section 6 details our experiments and discussion, followed by conclusions in Section 7.

2. RELATED STUDY

2.1. DCT-Embedding-Based Adaptive Steganography

Most steganographic systems aim to minimize the distortion of the original cover. A practical framework to minimize statistical detectability when designing undetectable steganography was recently presented [Filler and Fridrich 2011]. To design DCT-embedding-based adaptive steganography, an inter-/intrablock cost model was given, as well as the performance of embedding algorithms based on the inter-/intrablock cost model. The proposed DCT-embedding-based adaptive steganography was experimentally validated as being highly secure [Filler and Fridrich 2011]. In what follows, we briefly introduce this practical framework.

Minimal-distortion steganography can be implemented by minimizing the following cost function:

$$D(\mathrm{x}, \mathrm{y}) = \sum_{i=1}^{n} \rho_i(\mathrm{x}, y_i); \tag{1}$$

where $\rho_i(\mathrm{x}, y_i) \in \Re$ is the cost changing the ith cover pixel x_i to y_i.

To design DCT-embedding-based adaptive steganography, an inter-/intrablock cost model has been defined by Filler and Fridrich [2011]. Let $\theta = (\theta_{ir}, \theta_{ia}) \in \Re^{(2\Delta+1)+1} \times \Re^{(2\Delta+1)+1}$ be the model parameters describing the cost of disturbing inter- and intrablock dependencies with $\theta_{ir} = (\theta_{ir,-\Delta}, \ldots, \theta_{ir,\Delta}, \theta_{ir,\bullet})$ and $\theta_{ia} = (\theta_{ia,-\Delta}, \ldots, \theta_{ia,\Delta}, \theta_{ia,\bullet})$. The cost of changing any AC DCT coefficients x_{ij} to $y \in \mathrm{I}_{ij}\mathrm{I}_{ij} = \left\{x_{ij} - 1, x_{ij}, x_{ij} + 1\right\} \cap \mathrm{I}$ is given by

$$\rho_{ij}(\mathrm{x}, y) = \Theta(y) = \begin{cases} 0 & \text{if } y = x_{ij} \\ \infty & \text{if } y \notin \mathrm{I}_{ij} \\ \sum_{z \in \mathrm{N}_{ia}} \theta^2_{ia, x_{ij}-z} + \sum_{z \in \mathrm{N}_{ir}} \theta^2_{ir, x_{ij}-z} & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathrm{N_{ia}}$ and $\mathrm{N_{ir}}$ are intra- and interblock neighborhoods. Based on the inter-/intrablock cost model, the embedding algorithms are optimized by using the multilayered Syndrome-Trellis Codes [Filler et al. 2011] to minimize the L2R_L2LOSS criterion [Filler and Fridrich 2011], with SVM and CC-PEV feature set [Kodovsky and Fridrich 2009] and Cross-Domain Feature set [Kodovsky and Fridrich 2011], respectively. The experiments show that proposed DCT-embedding-based adaptive steganography has greatly improved the state of DCT-embedding-based steganography. More technical details are provided in the reference [Filler and Fridrich 2011].

2.2. YASS and a Detection Algorithm

The original YASS algorithm presented in Solanki et al. [2007] includes the following steps:

1) Repeat-Accumulate error correction code is used to encode the payload.
2) The cover image is divided into big blocks of T × T (T = 9, 10, . . . ,15), denoted by the B-block.
3) In each B-block, an 8 × 8 block is randomly selected for payload embedding.
4) The embedding includes the following operations:
   a) Selected 8 × 8 block is transformed using a two-dimensional DCT.
   b) The DCT coefficients are divided by a quantization table, corresponding to the hiding quality factor $\mathrm{QF_h}$.
   c) By employing the Quantization Index Modulation (QIM) strategy [Chen and Wornell 2001], binary hidden bits are embedded into the 19 low-frequency AC DCT coefficients whose values are nonzeros to enhance the robustness of the embedded data.
   d) The modified 8 × 8 block is transformed back to the spatial domain.
5) The modified image is encoded in JPEG format with the advertising quality factor QF_a.

As we presented in the reference [Liu 2011a], although YASS embedding is not confined to the 8 × 8 block of the final JPEG compression, the location of the embedding block in the B-block is not random enough. By using QIM-based embedding, YASS also introduces additional zero DCT coefficients in the modified 8 × 8 block, and hence, the following algorithm was designed to break YASS [Liu 2011a].

Zero-value density-based approach to steganalysis of YASS [Li et al. 2009]
Transform a JPEG image under examination to spatial domain, denoted by I_1*;*
For $T = 9$ *to* 15
  For $s = 1$ *to* T

(a) *Divide* I_s *into nonoverlapping consecutive* $T \times T$ *B-blocks;*
(b) *Collect* 8×8 *blocks from the upper left of all B-blocks and perform 2D DCT;*
(c) *Quantize the DCT coefficients by using* QF_a*;*

(*d*) *Compute the probability of zero rounded requantized DCT coefficients in candidate embedding bands and denote it by* $Z_T(s)$*;*
(*e*) *Crop the first s columns and the first s rows of* I_1 *to generate a new image* I_{s+1} *for the next inner-loop;*

End
Compute the values of $\frac{1}{T-7}\sum_{i=1}^{T-7} Z_T(i)$ *and* $\frac{1}{7}\sum_{j=T-6}^{T} Z_T(j)$ *as features.*
End

As shown by this algorithm, the features are extracted from the candidate blocks along the diagonal direction of B-blocks, rather than from all possible 8×8 candidate blocks in B-blocks. In a B-block with the size of $T \times T$, there are a total of $(T - 7) \times (T - 7)$ block candidates for embedding. Unfortunately, the previous algorithm only selects the $(T - 7)$ blocks along a diagonal direction, not all candidate blocks, and as a result, the chance of the candidates along the diagonal direction only hits $1/(T - 7)$. While the value of T is large, the hit ratio is fairly low. For instance, if $T = 15$, the hit ratio is only $1/8 = 0.125$. The experimental results shown by Li et al. [2009] also demonstrate that the detection accuracy is not satisfactory with a large T value.

2.3. Neighboring Joint Density-Based JPEG Steganalysis

In our previous work on a neighboring joint density-based approach to steganalysis of spatial domain-based steganographic system, the features were designed in the format of statistical correlation on neighboring joint density [Liu et al. 2005]. Inspired by a multivariate generalized Gaussian distribution (MGGD) model in the wavelet that was successfully used for image denoising [Cho and Bui 2005], we discussed the MGGD in the DCT domain and pointed out that approximate distribution of neighboring joint density of DCT coefficients may be modeled by MGGD, and information hiding generally affects the distribution [Liu et al. 2009a, 2011a]. Our study also shows that besides information hiding, JPEG-based double compression and interpolation modify the neighboring joint density also and hence leave a clue to reveal the manipulations [Liu et al. 2006, 2008a, 2009a, 2011b, 2011c, 2013; Qiao et al. 2013]. Our experimental results indicate that a neighboring joint density-based approach outperforms the Markov transition probability-based approach in JPEG steganalysis [Liu et al. 2009a, 2011a]. We analyzed the reason that the neighboring joint density-based approach is generally superior to the highly referenced Markov-based approach: "it is the modification of the neighboring joint density that results in the modification of Markov transition probability" [Liu et al. 2011a]. We can completely derive Markov transition probability from neighboring joint density, but we cannot derive the neighboring joint density from Markov transition probability; in other words, neighboring joint density contains more discriminant information compared to Markov transition probability.

Normally, neighboring joint density of DCT coefficients is symmetric to the origin. We have designed the neighboring joint density features on the absolute array of DCT coefficients [Liu et al. 2011a], described as follows.

2.3.1. Neighboring Joint Density on Intrablock. Let F denote the quantized DCT coefficient array consisting of $M \times N$ blocks $F_{ij}(i = 1, 2, \ldots, M; j = 1, 2, \ldots, N)$. The intrablock neighboring joint density matrix on horizontal direction $absNJ_{1h}$ and the matrix on vertical direction $absNJ_{1v}$ are given by

$$absNJ_{1h}(x, y) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{8}\sum_{n=1}^{7}\delta(|c_{ijmn}| = x, |c_{ijm(n+1)}| = y)}{56MN}. \tag{3}$$

$$absNJ_{1v}(x, y) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{7}\sum_{n=1}^{8}\delta(|c_{ijmn}| = x, |c_{ij(m+1)n}| = y)}{56MN}, \tag{4}$$

where c_{ijmn} is the DCT coefficient located at the mth row and the nth column in the block F_{ij}; $\delta = 1$ if its arguments are satisfied; otherwise $\delta = 0$; x and y are integers. For computational efficiency, we define $absNJ_1$ as the neighboring joint density features on intrablock, calculated as follows:

$$absNJ_1(x, y) = \left\{absNJ_{1h}(x, y) + absNJ_{1v}(x, y)\right\}/2. \tag{5}$$

In our prior detection, the values of x and y are in the range [0, 5], and $absNJ_1$ consists of 36 features.

2.3.2. Neighboring Joint Density on Interblock. The interblock neighboring joint density matrix on horizontal direction $\text{absNJ}_{2\text{h}}$ and the matrix on vertical direction $\text{absNJ}_{2\text{v}}$ are constructed as follows:

$$absNJ_{2h}(x, y) = \frac{\sum_{m=1}^{8}\sum_{n=1}^{8}\sum_{i=1}^{M}\sum_{j=1}^{N-1}\delta(|c_{ijmn}| = x, |c_{i(j+1)mn}| = y)}{64M(N-1)}. \tag{6}$$

$$absNJ_{2v}(x, y) = \frac{\sum_{m=1}^{8}\sum_{n=1}^{8}\sum_{i=1}^{M-1}\sum_{j=1}^{N}\delta(|c_{ijmn}| = x, |c_{(i+1)jmn}| = y)}{64(M-1)N}. \tag{7}$$

We define absNJ_2 as the neighboring joint density features on interblock, calculated as follows:

$$absNJ_2(x, y) = \{absNJ_{2h}(x, y) + absNJ_{2v}(x, y)\}/2. \tag{8}$$

Similarly, the values of x and y are in [0, 5] and absNJ_2 has 36 features. In our previous approach, the neighboring joint density features defined by Equations (5) and (8) are denoted by $absNJ$, containing 72 features.

3. CALIBRATED NEIGHBORING JOINT DENSITY-BASED STEGANALYSIS

We have shown and validated the modification of the neighboring joint density caused by information hiding of several DCT-embedding steganographic systems [Liu et al. 2011a]. Regarding DCT-embedding adaptive steganography that aims to minimize the distortion cost through Syndrome-Trellis Codes [Filler and Fridrich 2011], although the modification is very small, it does change the neighboring joint density (Figure 1).

Figure 1(a) shows a JPEG cover that was downloaded from the website [BOSS]. Figure 1(b) gives the JPEG steganogram produced by using the DCT-embedding-based adaptive hiding algorithm with the relative payload of 0.4 bits per nonzero AC (bpac). Figure 1(c) demonstrates the difference of the intrablock-based neighboring joint density when comparing the cover and the steganogram, and Figure 1(d) shows the difference of the neighboring joint density of the absolute array of DCT coefficients when comparing the cover and the steganogram.

It should be noted that we do not have the original cover as a reference while detecting steganography. For example, in Figure 1, only given the JPEG image (a) or (b), not both, we need to determine whether the image under examination is a cover or a steganogram; it is impossible for us to obtain the density difference shown in (c) and (d) in real detection. We should also mention that the neighboring joint density varies across different JPEG images. Therefore, there are still limitations to detecting the steganogram if we only adopt the neighboring joint density feature set without any reference, originally presented in Liu et al. [2009a, 2011a].

To capture the modification of the density caused by data embedding, suggested by the self-calibration that was presented in Fridrich [2004] and based on our previous steganalysis method [Liu et al. 2011a] and forgery detection method [Liu 2011b], we design a calibrated neighboring joint density-based approach, described as follows:

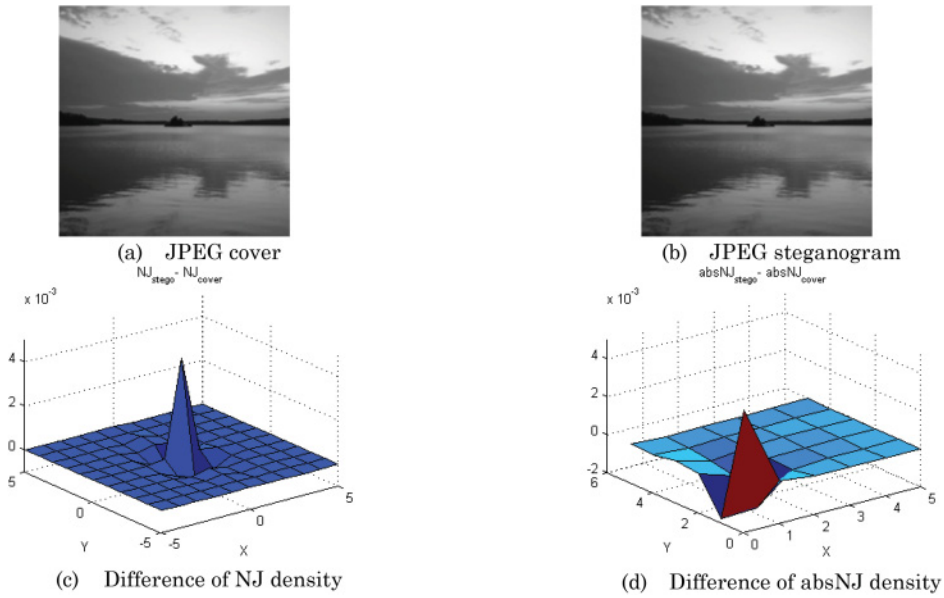

Fig. 1. An example to demonstrate the modification of neighboring joint (NJ) density features by DCT-embedding-based adaptive steganography.

1. The neighboring joint density features $absNJ_1(x, y)$ and $absNJ_2(x, y)$, defined by Equations (5) and (8), are extracted from a JPEG image under examination.
2. The testing JPEG image is decoded to spatial pixel values and cropped by i rows and j columns ($0 \le i < 7, 0 \le j < 7$, and $I + j > 0$). The cropped image is encoded in JPEG format with the same quantization matrix, and the joint density features denoted by $absNJ^c_{1i,j}(x, y)$ and $absNJ^c_{2i,j}(x, y)$ are extracted from the cropped and recompressed JPEG images.
3. The mean values of $absNJ^c_1(x, y)$ and $absNJ^c_2(x, y)$ are calculated by

$$\overline{absNJ^c_1}(x, y) = \frac{1}{63} \sum_{(i,j)} absNJ^c_{1i,j}(x, y) \tag{9}$$

$$\overline{absNJ^c_2}(x, y) = \frac{1}{63} \sum_{(i,j)} absNJ^c_{2i,j}(x, y). \tag{10}$$

4. The differential joint density features are given by

$$absNJ^D_1(x, y) = \overline{absNJ^c_1}(x, y) - absNJ_1(x, y) \tag{11}$$

$$absNJ^D_2(x, y) = \overline{absNJ^c_2}(x, y) - absNJ_2(x, y). \tag{12}$$

In our detection, we either adopt the neighboring joint density features, given by Equations (5) and (8), and the reference density features, given by Equations (9) and (10), together as a detector or adopt the features defined in Equations (5), (8), (11), and (12) together as a detector. We should note that both detectors are actually the same, because each one feature set can completely be derived from another. Our experiments also verify that both feature sets have approximately identical detection performance by using different classifiers. By using a Fisher linear discriminant and

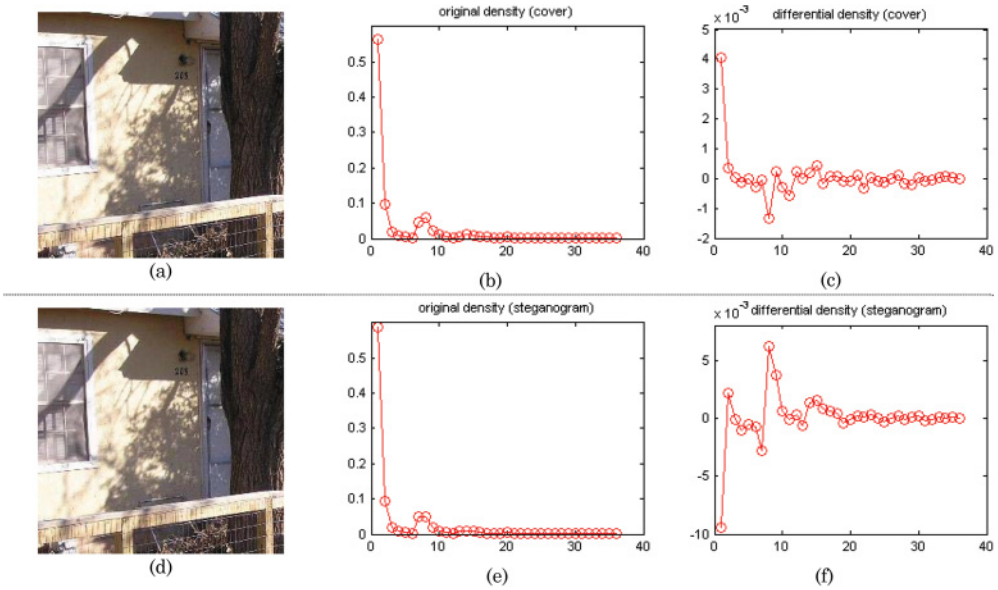

Fig. 2. A demonstration of a JPEG cover image (a) and the F5 steganogram (d). Original neighboring joint densities are shown in (b) and (e), and the self-differential densities are given in (c) and (f), respectively.

logistic regression classifier, especially, we have obtained exactly the same detection results. The detector of calibrated neighboring joint density containing the features by Equations (5), (8), (9), and (10) or by Equations (5), (8), (11), and (12) is denoted by CC-absNJ.

To demonstrate the effectiveness of a calibrated neighboring joint density-based approach, Figure 2(a) shows a JPEG cover image, Figure 2(b) plots the neighboring joint density defined in Equation (5), and Figure 2(c) manifests the differential joint density, defined in Equation (11). Figure 2(d) shows the JPEG steganogram produced by the F5 algorithm, Figure 2(e) is the neighboring joint density defined in Equation (5), and Figure 2(f) gives the differential joint density defined in Equation (11). The original neighboring joint density from cover and the density from steganogram are different, as are the differential joint densities.

Figure 3(a) shows a JPEG cover image and Figure 3(d) presents the steganogram produced by using the adaptive-embedding algorithm [Filler and Fridrich 2011]. Original neighboring joint densities from the cover and from the steganogram are given in (b) and (e), respectively, and the differential densities are plotted in (c) and (f), respectively. The difference of the self-differential density between the cover and the steganogram is noticeable. Figure 2(a) and Figure 3(a) also demonstrate that different JPEG images have different neighboring joint densities, implying the importance of self-differential density for steganalysis.

4. CALIBRATED NEIGHBORING JOINT DENSITY-BASED APPROACH TO YASS STEGANALYSIS

By searching all possible 8×8 candidate blocks in B-blocks, we extract the neighboring joint density of the DCT coefficients from all candidate blocks that are possibly used to carry hidden data and the 8×8 noncandidate block neighbors that are not used for information hiding and then calculate the difference in the joint density values of the

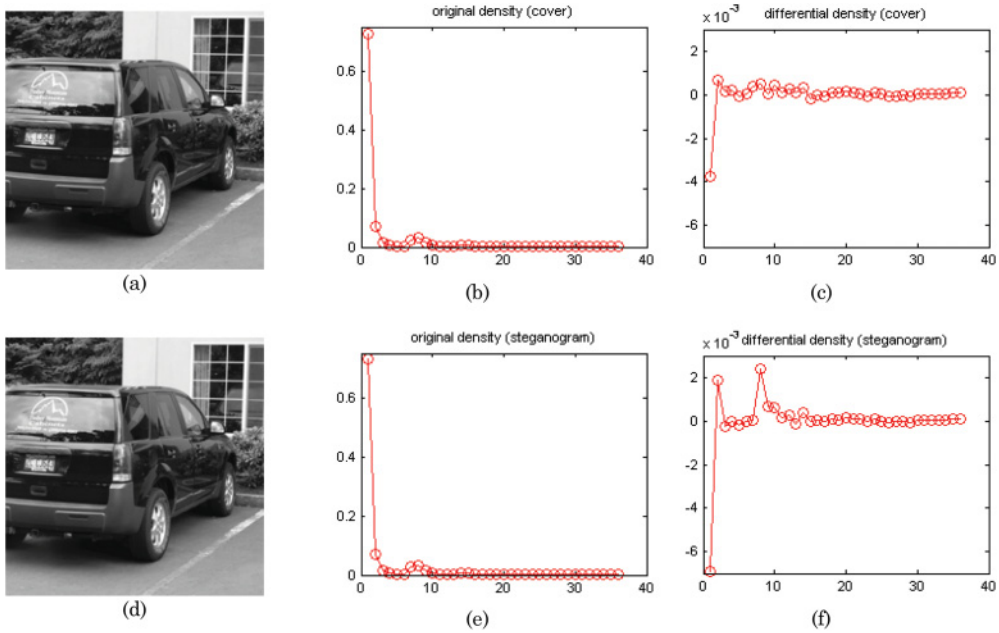

Fig. 3. A JPEG cover image (a) and the adaptive-embedding steganogram (d). Original neighboring joint densities are shown in (b) and (e), and the self-differential densities are given in (c) and (f), respectively.

candidates and the noncandidate neighbors. Our algorithm of feature design to detect YASS steganogram is described as follows:

Starting from the large B-block parameter T = 9:

1. Decode the JPEG image under scrutiny to spatial domain and divide it into nonoverlapping consecutive $T \times T$ B-blocks.
2. In each $T \times T$ B-block, search all 8 × 8 blocks possibly used for information hiding, in a total of $(T - 7)^2$ candidate blocks. The set of all candidate blocks of the image under detection is denoted by CB. For each candidate block CB(i)($I = 1, 2, \ldots, CN$; CN is the number of all candidate blocks on the testing image), subtract 128 from each pixel value, and then apply two-dimensional DCT transform, quantize the DCT coefficients by using the quantization matrix corresponding to QF_a, and obtain the absolute DCT coefficient array. The neighboring joint density features on the intrablock of CB(i), denoted by $absNJ(i; x, y)$, is given by

$$absNJ(i;x,y) = 0.5 \times \left(\frac{\sum_{m=1}^{8}\sum_{n=1}^{7}\delta\left(\left|c_{mn}^{i}\right| = x, \left|c_{m(n+1)}^{i}\right| = y\right)}{56} + \frac{\sum_{m=1}^{7}\sum_{n=1}^{8}\delta\left(\left|c_{mn}^{i}\right| = x, \left|c_{(m+1)n}^{i}\right| = y\right)}{56} \right), \tag{13}$$

   where c_{mn}^{i} is the DCT coefficient located at the mth row and the nth column in the candidate block CB(i); $\delta = 1$ if its arguments are satisfied; otherwise, $\delta = 0$; x and y are integers.
3. From all 8 × 8 blocks that are adjacent to the candidate block CB(i) in the horizontal/vertical direction but without any overlap to CB(i), the adjacent 8 × 8 blocks that do not belong to CB are denoted by NC(i,j). Generally, noncandidate 8 × 8 blocks

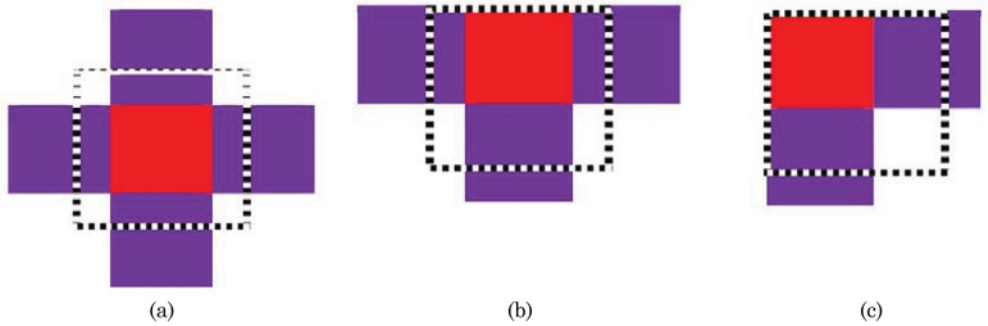

Fig. 4. A candidate block is located in a B-block (dashed), and the noncandidate neighbors are across two B-blocks.

must be across two adjacent T×T B-blocks, such as when a $T \times T$ B-block is not on the boundary or on the corner of an image under examination:

(a) If an 8 × 8 block candidate is located inside of the B-block without any overlap to the B-block boundary, it has four noncandidate neighbors, shown by Figure 4(a).
(b) If an 8 × 8 block candidate overlaps at only one of the four boundary borders of the B-block, it has three noncandidate neighbors, shown by Figure 4(b).
(c) If an 8 × 8 block candidate overlaps at two of the four boundary borders of the B-block or is located at one of the four corners of the B-block, it has two noncandidate neighbors, shown by Figure 4(c).

4. The neighboring joint density on the noncandidate neighboring block NC(i,j) is given by

$$absNJ(i,j;x,y) = 0.5 \times \left(\frac{\sum_{m=1}^{8}\sum_{n=1}^{7}\delta\left(\left|c_{mn}^{ij}\right| = x, \left|c_{m(n+1)}^{ij}\right| = y\right)}{56} + \frac{\sum_{m=1}^{7}\sum_{n=1}^{8}\delta\left(\left|c_{mn}^{ij}\right| = x, \left|c_{(m+1)n}^{ij}\right| = y\right)}{56} \right), \tag{14}$$

where c_{mn}^{ij} is the DCT coefficient located at the mth row and the nth column in the noncandidate block NC(i, j). $\delta = 1$ if its arguments are satisfied; otherwise, $\delta = 0$; x and y are integers.

5. The mean value of the differential neighboring joint density between candidate blocks and noncandidate blocks is given by

$$diff\text{-}absNJ(x,y) = \frac{\sum_i absNJ(i;x,y)}{count(CB)} - \frac{\sum_{(i,j)} absNJ(i,j;x,y)}{count(NC)}, \tag{15}$$

where count (CB) gives the total number of candidate blocks, and count (NC) gives the total number of noncandidate blocks on the testing image.

The features defined in Equation (15) constitute the feature set to detect the YASS steganogram produced by large B-block size T. The values of x and y are set from (0, 0), (0, 1), (0, 2), (1, 0), ... to (2, 2), in a total of nine differential neighboring joint density features for a single value of B-block size T.

6. While T < 16, set T + 1 to T, and repeat steps 1 to 6. The final detector contains 63 differential features for all possible T parameters (T = 9, 10, ... 15).

Figure 5 shows a cover and YASS steganograms produced with a B-block size of 9, 11, and 13 on the left. The diff-absNJ features extracted from the cover and the

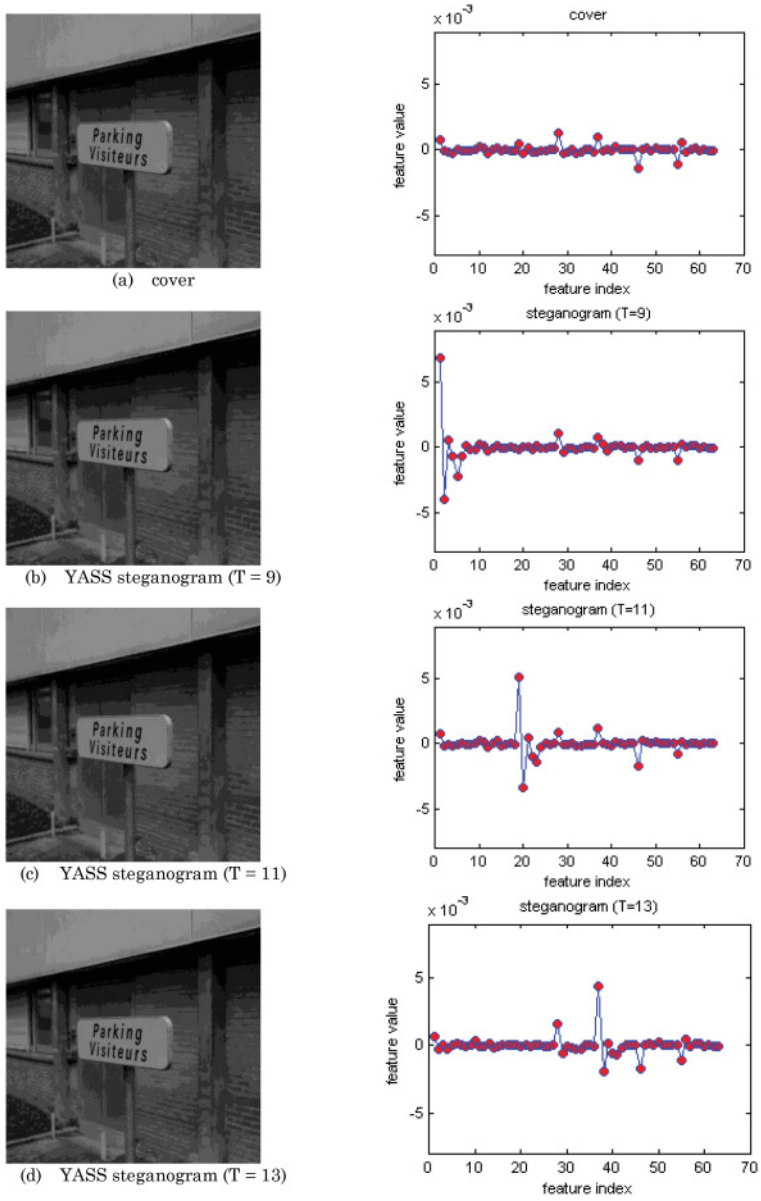

Fig. 5. Different patterns of diff-absNJ features among cover image and YASS steganograms ($QF_h = QF_a = 75$) with B-block parameter T = 9, 11, and 13. The cover and steganograms are shown on the left and the diff-absNJ features on the right.

steganograms are shown on the right, manifesting different patterns between the cover and different steganograms produced by different B-block size.

5. INTEGRATION OF CALIBRATED NEIGHBORING JOINT DENSITY AND SPATIAL RICH MODELS FOR SEAM-CARVED FORGERY DETECTION IN JPEG IMAGES

In seam carving, finding the seam is completed with the path of minimum cost from one end of the image to another. While seam carving allows for removal of selected whole

objects from photographs or removing/inserting some seams, the manipulation occurs in the spatial domain—it directly modifies the pixel values in the spatial domain. In addition to altering the pixel values, the removal or insertion of seams also results in the change of some pixel positions in the original image and in destroying the original compression block structure, hence leaving the trace of the manipulation both in the spatial domain and in the transform domain. Based on these facts, we use calibrated neighboring joint density features that have been described in Section 3 to reveal the modification in the transform domain. To keep track of the modification in the spatial domain, we directly make use of a spatial domain rich model, recently designed for steganalysis, to capture the modification of the statistical features [Fridrich and Kodovsky 2012]. We surmise that the spatial domain rich model may be very effective in detecting the seam-carving-based manipulation in the spatial domain since seam carving directly removes/inserts seams in the spatial domain and changes the pixel values and positions. In addition to the comparison of the detection performance of the calibrated neighboring joint density in the DCT domain and spatial rich-model-based features in the spatial domain, we integrate these two types of feature sets together for the detection with the expectation of obtaining better detection accuracy.

Figure 6 shows an example to verify the modification of the joint density in the DCT domain and the modification of the pixel values in grayscale format on the red, green, and blue channels. An untouched JPEG image and the forged JPEG image by seam carving are shown in (a) and (d), respectively. In the forgery, the image of the man at the center of the original photo has been removed. The neighboring joint densities in the DCT domain directly extracted from the untouched image and from the tampered image are given in (b) and (e), and the differential densities between original density and the calibrated density are given in (c) and (f) respectively. To reveal the modification in the spatial domain, Figure 6(g) gives the difference of the grayscale values between the tampering and the untouched photo, Figure 6(h). Figure 6(i) and Figure 6(j) demonstrate the difference of the pixel values on red, green, and blue channels, respectively; the tampering has been noticeably modified the pixel values. Some modifications go as high as 200, implying that the spatial-domain-based feature set could be very effective for the detection.

6. EXPERIMENTS

6.1. Steganalysis of DCT-Embedding-Based Nonadaptive and Adaptive Steganography

6.1.1. Datasets. The 5,000 original color TIFF raw format digital images used in the experiments are 24-bit, 640 $\times$ 480 pixels, lossless true color, never compressed. According to the method in Liu et al. [2008b, 2008c, 2009a, 2011a], we cropped these original images into 256 $\times$ 256 pixels in order to eliminate the low-complexity parts and converted the cropped images into JPEG format with the default quality. The following nonadaptive steganograms are generated with different hiding ratios, measured by relative payload, or the ratio of the number of DCT-coefficients modified to the total number of nonzero-valued AC DCT-coefficients.

1. **F5** – Westfeld proposed the algorithm F5 that withstands visual and statistical attacks yet still offers a large steganographic capacity [Westfeld 2001].
2. **Steghide** – Hetzl and Mutzel [2005] designed a graph-theoretic approach for information hiding based on the idea of exchanging rather than overwriting pixels. Their approach preserves first-order statistics, and the detection on the first order does not work.
3. **Model-based steganography without deblocking (MB1)** – Sallee [2003] presented an information-theoretic method for performing steganography. Using the model-based methodology, an example steganography method is proposed for JPEG

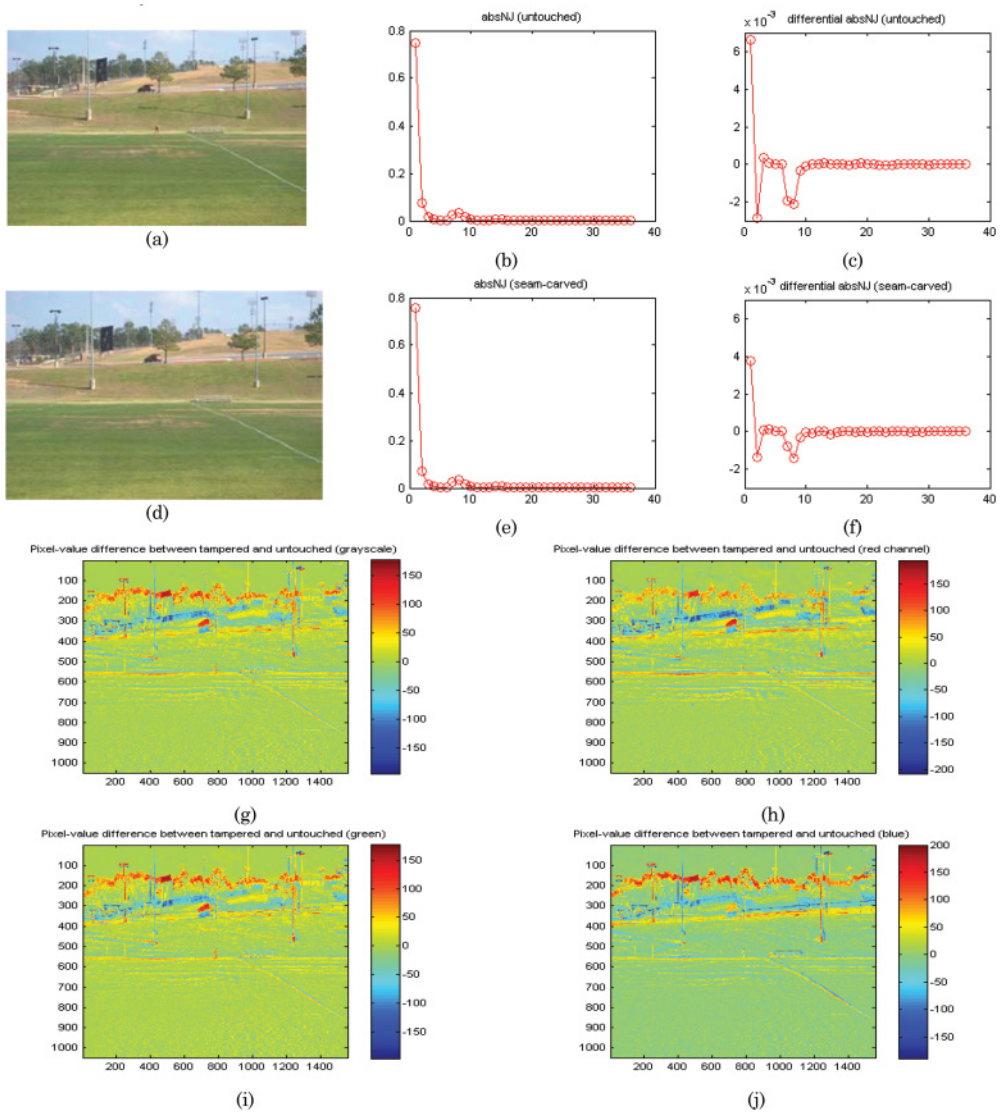

Fig. 6. Untouched JPEG image (a) and the forged image (d). Original neighboring joint densities in DCT domain are shown in (b) and (e), and the differential densities between original density and calibrated density are given in (c), and (f), respectively. The differences of the pixel values between the tampered (d) and untouched (a) are shown in (g) on the grayscale, (h) on the red channel, (i) on the green channel, and (j) on the blue channel.

images, which achieves a higher embedding efficiency and message capacity than previous methods while remaining secure against first-order statistical attacks.

4. **Model-based steganography with deblocking (MB2)** – Based on model-based steganography, Sallee [2005] presented a method to defend against a "blockiness" steganalysis attack.
5. **Adaptive steganography in JPEG images** – In order to produce DCT-embedding-based adaptive steganography, 1,000 BOSSRank cover images downloaded from the website [BOSS] are converted into JPEG images first at the quality

Table I. Compared Detectors

Detector	Feature Dimensionality	Reference
CC-absNJ	144	Features defined by Equations (5), (8), (11), and (12)
absNJ	72	Liu et al. [2011a]
CC-PEV	548	Kodovsky and Fridrich [2009]
PEV	274	Pevny and Fridrich [2007]
Markov	486	Chen and Shi [2008]
CC-C300	48,600	Kodovsky and Fridrich [2011]
CF	7,850	Kodovsky et al. [2012]
CC-JRM	22,150	Kodovsky and Fridrich [2012]
CC-JRM+SRMQ1	35,263	Fridrich and Kodovsky [2012]; Kodovsky and Fridrich [2012]

factor "75." The adaptive steganograms are produced by using the adaptive DCT-embedding hiding tool [Filler and Fridrich 2011], and the parameter of hiding bits per nonzero AC (bpac) is set from 0.1 to 0.35 with the step size of 0.05 bpac.

6.1.2. Detectors and Learning Classifiers. In our study, the following steganalysis detectors are compared: (1) 72-dimensional ***absNJ***, a neighboring joint density-based JPEG steganalysis originally designed in Liu et al. [2011a]; (2) 144-dimensional ***CC-absNJ***, a calibrated neighboring joint density, consisting of 144 features, defined by Equations (5), (8), (11), and (12), or by Equations (5), (8), (9), and (10) (we argue that both 144-dimensional feature sets are actually identical in terms of the detection capability); (3) 548-dimensional ***CC-PEV*** [Kodovsky and Fridrich 2009]; (4) 274-dimensional ***PEV*** [Pevny and Fridrich 2007]; (5) 486-dimensional **Markov**-process-based detector [Chen and Shi 2008]; (6) 48,600-dimensional rich model ***CC-C300***, a high-dimensional rich model for JPEG steganalysis [Kodovsky and Fridrich 2011]; (7) 7,850-dimensional compact rich model ***CF*** for JPEG steganalysis [Kodovsky et al. 2012]; (8) 22,510-dimensional Cartesian calibrated JPEG domain rich model ***CC-JRM*** [Kodovsky and Fridrich 2012]; and (9) a union of spatial domain rich model with the fixed quantization q = 1c, 12,753-dimensional *SRMQ1* [Fridrich and Kodovsky 2012], and 22,510-dimensional *CC-JRM*, denoted by ***CC-JRM+SRMQ1***, a total of 35,263 features [Kodovsky and Fridrich 2012]. Table I lists these feature sets.

Support Vector Machines (SVMs) [Change and Lin 2011; Vapnik 1998], Fisher's Linear Discriminant (FLD) to minimize the errors in the least square sense [Heijden et al. 2004], and an ensemble classifier that was used with rich models for steganalysis [Kodovsky et al. 2012] are employed in our comparison study. It should be noted that the computational cost by SVM is too high for rich models due to the high dimensionality of the feature set, and rich-model-based steganalysis detectors are not suitable with SVM. However, the low-dimensional detectors proposed in our study are easily utilized with SVM.

To select SVM for the low-dimensional detectors, we compare the popular algorithms LibSVM [Change and Lin 2011; Yan 2006], SVM_light [Jaochimes 2002], the SVM algorithms implemented in PRtools [Heijden et al. 2004], and five SVM learning algorithms in LIBLINEAR [Fan et al. 2008]. We compare these SVM algorithms with different parameters including linear, polynomial, and radial basis function (RBF) kernels. In our comparison, although the algorithms implemented in the LIBLINEAR package are the fastest, the accuracy is the lowest; LibSVM generally obtains the best detection accuracy. Therefore, we finally employed LibSVM with optimal kernel parameters after comparing different combinations of kernel parameters by grid search [Change and Lin 2011].

While we apply the ensemble classifier that was used in Kodovsky et al. [2012], the optimized parameters are computed first, including the optimization of the subdimensionality and optimal base learning classifiers. By optimizing the parameters and applying an optimized ensemble classifier to rich-model-based detectors, the computational cost is much higher than if using Fisher linear discriminant.

We perform 100 experiments for each feature set at each hiding ratio by using each classifier. In each experiment, 70% of the samples are randomly selected for training, and the other 30% samples are used for testing. The prediction outcomes on testing data can be divided into True Negative (TN), False Negative (FN), False Positive (FP), and True Positive (TP). Without losing generality, our detection accuracy is calculated by $0.5 * \text{TN}/(\text{TN}+\text{FP}) + 0.5 * \text{TP}/(\text{TP} + \text{FN})$.

6.1.3. Experimental Results. Tables IIA to IIE list the mean values of detection accuracy over 100 experiments to detect F5, steghide, MB1, MB2, and adaptive steganography in JPEG image, respectively. In the results, by applying each learning classifier to the nine detectors, the best testing accuracy is highlighted in bold; by applying the three learning classifiers to the nine detectors, the best testing accuracies in bold are squared.

In detecting F5 steganography, calibrated neighboring joint density is generally superior to the other eight detectors, shown by Table IIA. The best testing accuracy is obtained by CC-absNJ with LibSVM. In detecting steghide steganography, CC-absNJ and the union of CC-JRM and SRMQ1 generally outperform the other seven detectors, shown by Table IIB. In detecting MB1 and MB2 steganography, shown by Tables IIC and IID, a calibrated neighboring joint density-based detector (CC-absNJ) obtains the best detection accuracy. In adaptive steganalysis, by using an ensemble classifier, the union of CC-JRM and SRMQ1 performs the best in detecting the steganograms at a relative payload of 0.1 bpac, with the testing accuracy of 85.7%. The application of an ensemble classifier to another rich model detector (CC-C300) cannot obtain optimal base learning classifiers; the detection is not available at relative payload 0.1 bpac. While detecting adaptive steganograms at 0.15 bpac to 0.35 bpac, CC-absNJ is comparable to the union of CC-JRM and SRMQ1, delivering better detection accuracy than the other seven detectors.

6.2. Steganalysis of YASS

6.2.1. Setup. The original 1,000 BOSSRank cover images downloaded from the website [BOSS] are used for YASS embedding. We set $QF_h = QF_a = 75$ in production of the steganograms. Accordingly, we encode the 1,000 BOSSRank cover images in JPEG format at the quality factor of 75 as cover images. To create YASS steganograms, QF_h and QF_a are set to the same quantization factor in order to avoid double JPEG compression in YASS steganograms. If QF_h is not equal to QF_a, the YASS steganograms could be detected by exposing the double JPEG compression. Double JPEG compression has been documented with very good detection performance [Chen and Hsu 2011; Liu et al. 2011c, 2013; Pevny and Fridrich 2007]. Additionally, the big B-block size T is set from 9 to 15 respectively to produce the steganograms.

To conduct a comparative study, we extract the *diff-absNJ* features defined in Equation (15), and the zero-value density features designed by Li et al. [2009]. LibSVM and FLD classifiers are used for classification. In each experiment, 50% of the samples are randomly selected for training, and the other 50% of samples are used for testing; 200 experiments are operated for each feature set at each B-block size by using each learning classifier for binary classification, and 200 experiments are conducted for each feature set by mixing covers and all YASS steganograms together for multiple-class classification. Our approach and zero-value density-based detection are based on the

TABLE IIA. F5 Steganalysis
The mean detection accuracy (%) over 100 experiments with LibSVM (**S**), Fisher linear discriminant (**F**), and ensemble classifier (**E**).

	Relative Payload																				
	0.051			0.077			0.105			0.137			0.185			0.282			0.354		
Detector	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E
CC-absNJ	**94.4**	**94.6**	**92.0**	**96.6**	**95.4**	**94.1**	**98.4**	**97.1**	**97.4**	**99.0**	**98.2**	**98.5**	**99.3**	98.9	**99.1**	**99.9**	**99.6**	99.4	**100**	99.6	99.7
absNJ	91.9	91.0	86.7	93.9	91.1	89.7	96.5	92.9	95.0	97.6	94.7	96.0	97.9	95.8	97.0	99.2	98.7	97.2	99.8	99.7	99.6
CC-PEV	81.0	90.4	85.6	85.3	92.0	89.2	91.7	96.5	95.2	94.0	97.7	97.0	97.3	**99.3**	98.7	98.9	**99.6**	**99.6**	99.8	99.8	**99.9**
PEV	85.6	86.4	81.0	90.0	88.6	84.7	94.5	94.3	92.9	96.5	96.6	95.8	97.9	98.9	98.3	99.1	99.5	99.2	99.8	99.8	**99.9**
Markov	68.1	75.7	69.5	66.9	76.6	74.1	75.5	85.2	82.7	76.6	91.6	88.7	86.9	96.5	94.6	95.0	97.7	96.7	97.7	99.4	99.3
CC-C300	x	81.4	74.9	x	90.3	86.2	x	94.8	93.7	x	96.8	96.2	x	98.8	98.7	x	99.2	99.0	x	99.7	99.8
CF	x	77.0	78.4	x	87.0	85.2	x	88.5	93.4	x	89.4	96.7	x	92.8	98.6	x	95.3	99.1	x	99.0	99.8
CC-JRM	x	79.3	87.7	x	91.6	92.1	x	93.6	95.9	x	95.0	97.6	x	96.9	98.0	x	98.8	99.4	x	99.8	99.8
CC-JRM+SRMQ1	x	88.2	82.0	x	93.6	88.4	x	95.4	95.3	x	97.0	97.9	x	97.9	98.0	x	99.3	99.4	x	**99.9**	**99.9**

TABLE IIB. Steghide Steganalysis
The mean detection accuracy (%) over 100 experiments with LibSVM (**S**), Fisher linear discriminant (**F**), and ensemble classifier (**E**).

	Relative Payload																				
	0.021			0.029			0.036			0.044			0.055			10.073			0.114		
Detector	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E
CC-absNJ	**92.1**	**92.5**	87.4	**95.4**	**95.9**	93.3	**98.1**	**97.4**	96.9	**99.2**	98.3	98.7	**99.7**	99.2	99.4	**99.9**	**99.7**	**99.9**	**100**	**99.9**	**100**
absNJ	88.9	88.1	80.8	92.0	91.0	86.8	95.0	93.4	92.0	97.3	95.5	95.8	98.1	96.3	96.8	99.4	98.1	98.8	99.8	99.3	99.7
CC-PEV	82.4	89.5	83.7	83.9	93.0	89.2	90.1	96.8	94.5	94.2	**98.7**	97.5	96.7	**99.3**	98.4	99.1	**99.7**	99.5	99.7	99.8	99.8
PEV	82.4	82.6	74.4	85.5	85.7	80.9	90.5	92.1	89.0	95.3	96.4	94.5	97.5	97.9	96.7	99.3	99.5	99.1	99.7	99.7	99.7
Markov	72.9	83.3	77.4	75.0	85.9	81.9	84.1	91.7	89.3	89.6	96.0	94.0	93.7	97.4	96.2	97.9	99.0	98.5	99.2	99.4	99.3
CC-C300	x	74.8	65.6	x	81.5	72.1	x	87.4	79.8	x	91.5	85.5	x	95.2	91.6	x	97.8	96.9	x	98.9	98.7
CF	x	78.2	66.6	x	83.3	70.1	x	88.0	76.3	x	91.6	83.9	x	93.4	90.8	x	95.8	96.3	x	94.9	99.0
CC-JRM	x	85.1	75.3	x	90.4	84.2	x	94.3	91.0	x	96.5	95.5	x	97.8	97.5	x	98.9	99.2	x	98.5	99.7
CC-JRM+SRMQ1	x	85.9	**92.4**	x	91.2	**96.8**	x	95.2	**98.8**	x	97.3	**99.5**	x	98.5	**99.7**	x	99.4	99.8	x	99.2	99.9

TABLE IIC. MB1 Steganalysis
The mean detection accuracy (%) over 100 experiments with LibSVM (**S**), Fisher linear discriminant (**F**), and ensemble classifier (**E**).

	Relative Payload																				
	0.073			0.089			0.094			0.125			0.172			0.183			0.195		
Detector	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E
CC-absNJ	**99.5**	**98.1**	**98.0**	**99.9**	**99.7**	**99.8**	**99.7**	**99.5**	**99.5**	**99.9**	**98.6**	**98.6**	**99.9**	99.7	**99.9**	**100**	**99.8**	**100**	**100**	**99.9**	**100**
absNJ	95.8	94.8	91.5	97.4	95.3	95.9	97.9	96.5	96.9	98.4	94.6	96.8	99.8	98.9	99.7	99.8	99.5	99.8	99.9	99.4	99.9
CC-PEV	93.9	96.1	93.7	95.5	98.5	97.5	95.5	98.4	97.7	96.1	97.5	96.2	99.6	**99.8**	99.7	99.5	**99.8**	99.8	98.9	**99.9**	99.8
PEV	94.2	92.2	90.2	96.0	95.6	93.6	95.6	95.1	93.6	95.5	93.8	91.8	99.6	99.3	99.3	99.7	99.4	99.5	99.8	99.3	99.4
Markov	90.8	92.0	89.3	90.5	94.5	93.0	92.2	95.0	93.6	90.3	93.4	90.9	99.1	99.3	99.0	99.3	99.3	99.2	97.8	99.4	99.3
CC-C300	x	74.6	73.8	x	87.7	61.3	x	83.3	54.4	x	77.8	67.0	x	96.5	87.5	x	94.9	77.4	x	90.7	66.6
CF	x	57.0	88.9	x	91.4	93.1	x	89.6	86.3	x	85.7	82.7	x	97.7	98.9	x	98.3	98.8	x	96.7	96.1
CC-JRM	x	60.6	91.2	x	96.2	97.5	x	95.1	97.2	x	92.7	96.2	x	99.4	99.8	x	99.7	99.9	x	99.4	99.8
CC-JRM+SRMQ1	x	64.4	92.7	x	97.1	95.3	x	96.1	94.9	x	94.3	92.9	x	99.5	99.8	x	99.8	99.8	x	99.6	99.6

TABLE IID. MB2 Steganalysis
The mean detection accuracy (%) of over 100 experiments with LibSVM (**S**), Fisher linear discriminant (**F**), and ensemble classifier (**E**).

	Relative Payload																				
	0.101			0.120			0.131			0.168			0.226			0.245			0.271		
Detector	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E
CC-absNJ	**98.5**	96.4	**95.5**	**99.3**	98.4	**98.4**	**99.7**	99.1	**99.4**	**99.8**	99.1	**99.2**	**100**	99.7	99.8	**100**	**99.9**	**99.9**	**100**	**99.9**	**99.9**
absNJ	96.6	92.2	93.2	98.0	95.9	96.5	99.0	97.4	97.8	99.4	97.6	98.4	**100**	99.2	99.7	99.9	99.8	**99.9**	**100**	99.8	**99.9**
CC-PEV	95.0	**96.7**	95.4	95.5	**98.9**	98.1	97.0	**99.3**	99.1	98.9	**99.5**	**99.2**	99.7	**99.9**	**99.9**	99.6	**99.9**	**99.9**	99.8	**99.9**	**99.9**
PEV	94.0	92.3	90.6	96.2	95.8	94.4	98.0	97.8	97.4	99.2	98.9	98.6	99.9	99.8	99.7	99.8	99.7	99.7	99.9	**99.9**	**99.9**
Markov	90.7	92.0	89.9	87.2	94.7	92.6	92.4	96.5	95.2	96.4	97.1	96.1	98.3	99.2	98.9	98.5	99.5	99.3	99.3	99.7	99.7
CC-C300	x	68.9	63.9	x	84.9	56.8	x	90.2	66.5	x	95.5	79.7	x	96.9	80.8	x	95.7	78.2	x	97.5	89.4
CF	x	56.8	83.9	x	89.3	86.2	x	92.6	92.1	x	93.4	96.4	x	97.3	98.0	x	98.0	98.4	x	99.0	99.2
CC-JRM	x	60.8	90.4	x	94.5	95.7	x	96.1	97.5	x	97.0	98.2	x	99.2	99.7	x	99.5	99.8	x	99.6	**99.9**
CC-JRM+SRMQ1	x	63.3	90.4	x	95.4	94.5	x	97.1	96.6	x	97.8	98.1	x	99.4	99.6	x	99.6	99.8	x	99.7	**99.9**

TABLE IIE. Adaptive Steganalysis
The mean detection accuracy (%) over 100 experiments with LibSVM (**S**), Fisher linear discriminant (**F**), and ensemble classifier (**E**).

	Relative Payload																	
	0.1			0.15			0.2			0.25			0.3			0.35		
Detector	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E
CC-absNJ	**77.8**	78.0	78.3	**89.9**	89.5	89.9	**95.7**	**95.1**	95.4	**98.6**	**97.6**	98.1	**99.3**	**98.5**	99.0	**99.6**	99.0	99.5
absNJ	69.6	71.2	70.8	81.5	83.6	83.3	89.2	90.9	90.4	93.7	94.7	94.7	96.0	96.7	96.9	97.9	97.8	98.3
CC-PEV	58.0	66.6	70.0	68.8	82.0	83.1	76.5	90.6	90.9	84.4	96.0	96.0	89.5	97.6	97.8	94.7	99.0	98.9
PEV	66.0	64.5	65.6	77.7	78.0	78.6	86.3	87.7	87.6	92.9	94.2	94.0	95.8	96.7	96.7	98.1	98.8	98.7
Markov	50.1	51.5	50.9	53.3	66.6	67.1	57.5	77.8	78.8	65.8	85.7	87.4	69.4	91.4	92.4	73.5	94.8	95.3
CC-C300	x	82.0	NA*	x	89.5	63.8	x	93.6	84.8	x	96.5	93.3	x	97.8	96.3	x	98.6	98.0
CF	x	81.0	83.6	x	87.6	**90.5**	x	91.9	94.5	x	95.1	97.0	x	96.3	98.0	x	97.9	98.9
CC-JRM	x	81.1	81.9	x	88.5	89.8	x	92.5	94.0	x	95.8	96.7	x	97.1	97.9	x	98.5	98.7
CC-JRM+SRMQ1	x	**83.8**	**85.7**	x	**91.1**	88.6	x	94.8	**95.7**	x	97.0	**98.3**	x	98.1	**99.2**	x	**99.2**	**99.6**

*The testing result "NA" was caused by failure of the ensemble classifier without the final optimal base learner. "X" means that the classifier is not suitable for the feature set.

TABLE IIIA. Mean Testing Accuracy (%) over 200 Experiments (Binary-Class Classification)

		diff-absNJ		Zero-Value Density**		CC-JRM+SRMQ1***	
Noused*	B-Block, T	LibSVM	FLD	LibSVM	FLD	Ensemble	FLD
19	9	99.6	99.5	**99.8**	99.3	93.9	94.2
	10	**99.3**	**99.3**	98.8	98.9	95.7	94.5
	11	98.4	**98.5**	93.7	97.7	86.7	88.2
	12	96.8	**97.6**	74.3	94.3	86.8	87.1
	13	96.0	**96.4**	61.5	86.6	75.3	79.9
	14	93.6	**94.1**	53.0	77.1	71.8	76.5
	15	89.6	**90.1**	48.3	69.9	63.3	71.7
11	9	98.3	98.4	**99.7**	98.3	86.4	91.3
	10	97.3	97.9	95.2	**98.3**	90.4	91.3
	11	95.6	96.3	80.4	**96.4**	76.3	85.7
	12	93.4	**94.3**	67.5	90.3	76.5	83.1
	13	90.7	**91.5**	62.3	84.7	62.0	76.1
	14	86.3	**86.7**	60.4	75.0	59.4	73.7
	15	81.7	**82.1**	58.8	68.5	51.7	72.3
9	9	95.9	96.1	**99.3**	98.6	74.9	84.3
	10	93.6	93.7	91.8	**98.6**	74.8	82.6
	11	90.2	90.6	83.6	**96.0**	62.6	78.5
	12	87.3	87.1	73.5	**91.3**	62.2	76.5
	13	82.3	82.7	69.5	**86.7**	50.5	71.4
	14	76.8	76.9	67.2	**80.1**	NA	67.9
	15	72.7	73.1	66.1	**74.7**	NA	66.0

*_noused_ is the parameter to set the number of the first few AC DCT coefficients for data embedding in the block.
**Zero-value density-based approach assumes prior knowledge of the exact embedding position of the first few AC DCT coefficients in zigzag order for data embedding, which is generally inapplicable and not assumed by our approach and rich-model-based detection.
***When applying an ensemble classifier [Kodovsky et al. 2012] to the rich-model-based approach, the testing result "NA" means not available and is caused by the failure of the ensemble classifier, while the final optimal base classifier may not be generated.

exposure of potential candidate blocks for data hiding. Unlike the zero-value density-based approach, our method does not assume the embedding position on the first few positions in the candidate blocks. By using an ensemble classifier and FLD, we also employ the union of CC-JRM and SRMQ1 [Fridrich and Kodovsky 2012; Kodovsky and Fridrich 2012], a 35,263-deminsional feature set to detect steganograms without exposing the candidate blocks that are used for embedding.

6.2.2. Experimental Results. In binary classification, testing accuracy is measured by $0.5 * TP/(TP + FN) + 0.5 * TN/(TN + FN)$. The mean testing accuracy values over 200 experiments are given in Table IIIA. While the parameter *noused* is set to 19/14 while generating YASS steganograms, our method is generally more accurate than the other two compared methods. The zero-value density-based detection method [Li et al. 2009] performs well when detecting the YASS steganograms that were produced with a small B-block parameter; however, the detection performance apparently deteriorates while the parameter of the B-block size increases. The experimental results are consistent with the results in the reference [Li et al. 2009] and also validate our previous surmise.

In multiple-class classification, Tables IIIB, IIIC, and IIID give the confusion matrix with the mean testing results over 200 times. While the parameter of *noused* is 19, detector of zero-value density hits the correct detection of 16.2% for covers, and 84.4%, 73.4%, and 61.4% for YASS steganograms produced by large B-block size 13, 14, and 15 respectively. Our approach obtains correct detection results of 80.1%, 95.9%, 93.8%, and 90.3%, gaining considerable improvement. While the parameter of *noused* is getting

TABLE IIIB. Confusion Matrix of Mean Testing Accuracy (%) over 200 Experiments with LibSVM (Multiple-Class Classification, *noused* = 19)

Truth		Prediction Accuracy, %															
		Zero-Value Density								Diff-absNJ							
		Steganogram								Steganogram							
		cover	T=9	T=10	T=11	T=12	T=13	T=14	T=15	cover	T=9	T=10	T=11	T=12	T=13	T=14	T=15
Cover		**16.2**	0	0.9	5.1	8.4	18.2	23.9	27.3	**80.1**	0.0	0.1	0.5	1.7	2.4	5.4	9.9
	T=9	0.0	**100.0**	0	0	0	0	0.0	0	0.1	**99.8**	0	0.0	0.0	0.0	0.0	0
	T=10	0.4	0	**99.3**	0	0	0.0	0.2	0.0	0.4	0	**99.5**	0.0	0	0.0	0.1	0.0
	T=11	1.5	0	0	**97.1**	0.1	0.4	0.5	0.4	1.3	0.0	0.0	**98.1**	0.1	0.0	0.1	0.3
steg	T=12	2.2	0	0.0	0.1	**94.6**	0.7	1.1	1.3	2.0	0.0	0.0	0.1	**97.3**	0.2	0.3	0.3
	T=13	4.2	0	0.1	0.5	2.4	**84.4**	4.6	3.7	3.1	0.0	0.0	0.1	0.2	**95.9**	0.3	0.4
	T=14	5.1	0	0.1	1.5	3.2	6.2	**73.4**	10.1	4.3	0.0	0.0	0.1	0.2	0.2	**93.8**	1.3
	T=15	7.1	0	0.6	1.4	5.2	9.3	15.0	**61.4**	7.6	0.0	0.0	0.1	0.5	0.5	1.0	**90.3**

TABLE IIIC. Confusion Matrix of Mean Testing Accuracy (%) over 200 Experiments with LibSVM (Multiple-Class Classification, *noused* = 14)

Truth		Prediction Accuracy, %															
		Zero-Value Density								Diff-absNJ							
		Steganogram								Steganogram							
		cover	T=9	T=10	T=11	T=12	T=13	T=14	T=15	cover	T=9	T=10	T=11	T=12	T=13	T=14	T=15
Cover		**17.1**	0	1.3	6.6	13.0	17.9	21.3	23.0	**52.9**	0.4	1.1	2.0	4.6	7.4	14.0	18.4
	T=9	0.0	**99.9**	0	0	0	0	0	0.1	0.7	**98.4**	0	0.1	0.2	0.2	0.2	0.3
	T=10	0.9	0	**96.9**	0.0	0.3	0.2	0.8	0.8	1.2	0.0	**97.2**	0.1	0.2	0.2	0.5	0.6
	T=11	1.4	0	0.2	**93.2**	0.6	1.2	1.8	1.7	2.1	0.6	0.2	**95.2**	0.4	0.1	0.7	1.1
steg	T=12	2.3	0	0.6	0.5	**87.9**	2.5	3.1	3.2	3.6	0.2	0.2	0.4	**92.1**	0.9	1.1	1.4
	T=13	3.2	0	0.8	1.7	4.3	**75.9**	8.1	6.0	4.9	0.1	0.2	0.3	1.0	**89.8**	1.8	1.9
	T=14	5.5	0.0	1.8	3.9	6.0	12.1	**59.9**	10.9	6.9	1.8	0.6	0.4	0.6	1.7	**85.1**	4.5
	T=15	7.4	0.1	1.7	4.8	8.1	15.0	20.3	**42.7**	9.8	0.3	0.4	0.7	1.1	2.4	3.9	**81.5**

TABLE IIID. Confusion Matrix of Mean Testing Accuracy (%) over 200 Experiments with LibSVM (Multiple-Class Classification, *noused* = 9)

Truth		Prediction Accuracy, %															
		Zero-Value Density								Diff-absNJ							
		Steganogram								Steganogram							
		cover	T=9	T=10	T=11	T=12	T=13	T=14	T=15	cover	T=9	T=10	T=11	T=12	T=13	T=14	T=15
Cover		**38.0**	0	0.9	5.5	10.8	12.8	15.1	16.9	**19.7**	1.4	3.0	6.4	10.3	14.6	20.2	24.4
	T=9	0.2	**99.6**	0	0	0.0	0	0.0	0.1	1.0	**95.6**	2.9	2.6	3.5	6.4	8.8	9.6
	T=10	1.2	0.0	**93.3**	0.4	0.6	1.6	1.9	1.1	1.3	3.2	**91.6**	0.6	0.8	1.1	1.9	2.3
	T=11	1.1	0	0.4	**89.2**	1.3	2.4	4.2	1.5	2.3	0.5	0.7	**88.1**	0.9	1.8	2.4	3.3
steg	T=12	3.8	0	1.2	1.8	**82.8**	4.1	4.1	2.2	3.3	0.6	1.2	1.3	**84.1**	2.3	3.0	4.3
	T=13	4.8	0.0	1.1	3.9	5.7	**70.2**	8.4	6.0	4.1	0.7	1.0	1.5	2.5	**79.7**	4.9	5.6
	T=14	6.7	0.0	3.4	6.5	7.7	12.7	**53.8**	9.2	5.5	1.0	2.0	2.1	3.2	4.7	**72.7**	8.9
	T=15	11.0	0.2	3.3	7.2	10.5	16.4	19.8	**31.6**	6.7	0.9	1.6	2.6	4.1	6.7	9.0	**68.4**

smaller, the detection performance of our detection method decreases. On average, our approach is better than zero-value density.

While the parameter *noused* is set to a small value, for example, *noused* = 9, only the first nine AC DCT coefficients in zigzag order are used to generate steganograms. The zero-value density-based detection [Li et al. 2009] outperforms our detection and rich-model-based approach. We notice that zero-value density-based detection assumes prior knowledge of exact positions of the first few AC DCT coefficients that are used for

TABLE IIIE. Mean Testing Accuracy (%) over 100 Experiments in Detecting YASS (*noused* = 2)

		diff-absNJ$_{\text{lowfreq}}$**		diff-absNJ		Zero-value density*		CC-JRM+SRMQ1***	
Noused	B-Block, T	LibSVM	FLD	LibSVM	FLD	LibSVM	FLD	Ensemble	FLD
2	9	88.0	87.8	79.9	80.0	**97.6**	95.6	57.0	59.1
	10	84.3	83.6	76.9	77.0	**88.2**	87.9	52.9	55.5
	11	**82.1**	81.2	73.7	73.7	**82.1**	81.8	47.0	52.2
	12	**80.5**	78.9	69.8	69.8	75.1	77.2	NA	49.2
	13	**74.6**	73.6	65.7	66.1	72.3	73.6	NA	45.6
	14	**72.5**	71.9	62.6	62.9	65.2	68.3	NA	44.2
	15	**70.2**	69.7	59.7	60.0	60.4	64.4	NA	41.4

*A zero-value density-based approach assumes a prior knowledge of the exact embedding position of the first few AC DCT coefficients in zigzag order for data embedding, which is not assumed by diff-absNJ and CC-JRM+SRMQ1 detectors.
**diff-absNJ$_{\text{lowfreq}}$ approximately assumes the scope of embedding position of the DCT coefficients.
***When applying an ensemble classifier [39] to the rich-model-based feature set, the testing result "NA" means *not available* and is caused by the failure of the ensemble classifier, while the final optimal base classifier may not be generated. When applying FLD to the rich-model-based feature set, the testing accuracy is under 50%, while T > 12 due to overfitting.

data hiding; such prior knowledge is not assumed in our approach and rich-model-based approach.

To improve the detection performance of our approach while detecting the YASS steganograms produced by using a very small parameter of *noused* (e.g., *noused* = 2), we may modify Equations (13) and (14) in Section 4 as follows:

The neighboring joint density features in low frequency of candidate block CB(i), denoted by $absNJ(i;x,y)_{\text{lowfreq}}$, is given by

$$absNJ(i;x,y)_{lowfreq} = 0.5 \times \left(\frac{\sum_{m=1}^{3}\sum_{n=1}^{2}\delta\left(\left|c^{i}_{mn}\right|=x,\left|c^{i}_{m(n+1)}\right|=y\right)}{6} + \frac{\sum_{m=1}^{2}\sum_{n=1}^{3}\delta\left(\left|c^{i}_{mn}\right|=x,\left|c^{i}_{(m+1)n}\right|=y\right)}{6} \right). \quad (16)$$

The neighboring joint density in low frequency of noncandidate block NC(i,j), denoted by $absNJ(i,j;x,y)_{\text{lowfreq}}$, is given by

$$absNJ(i,j;x,y)_{lowfreq} = 0.5 \times \left(\frac{\sum_{m=1}^{3}\sum_{n=1}^{2}\delta\left(\left|c^{ij}_{mn}\right|=x,\left|c^{ij}_{m(n+1)}\right|=y\right)}{6} + \frac{\sum_{m=1}^{2}\sum_{n=1}^{3}\delta\left(\left|c^{ij}_{mn}\right|=x,\left|c^{ij}_{(m+1)n}\right|=y\right)}{6} \right). \quad (17)$$

The mean value of the differential neighboring joint density in low frequency between candidate blocks and noncandidate blocks is given by

$$diff\text{-}absNJ(x,y)_{lowfreq} = \frac{\sum_{i} absNJ(i;x,y)_{lowfreq}}{count(CB)} - \frac{\sum_{(i,j)} absNJ(i,j;x,y)_{lowfreq}}{count(NC)}. \quad (18)$$

Table IIIE gives the mean testing accuracy over 100 experiments. The results show that zero-value density-based detection outperforms others at low parameter of B-block (T = 9, 10) since it exactly assumes the embedding position in candidate block. However, due to the flaw of the original algorithm that does not search through all possible candidate blocks, while the B-block is large, for example, T > 12, the detection performance dramatically deteriorates. Compared to diff-absNJ, diff-absNJ$_{lowfreq}$

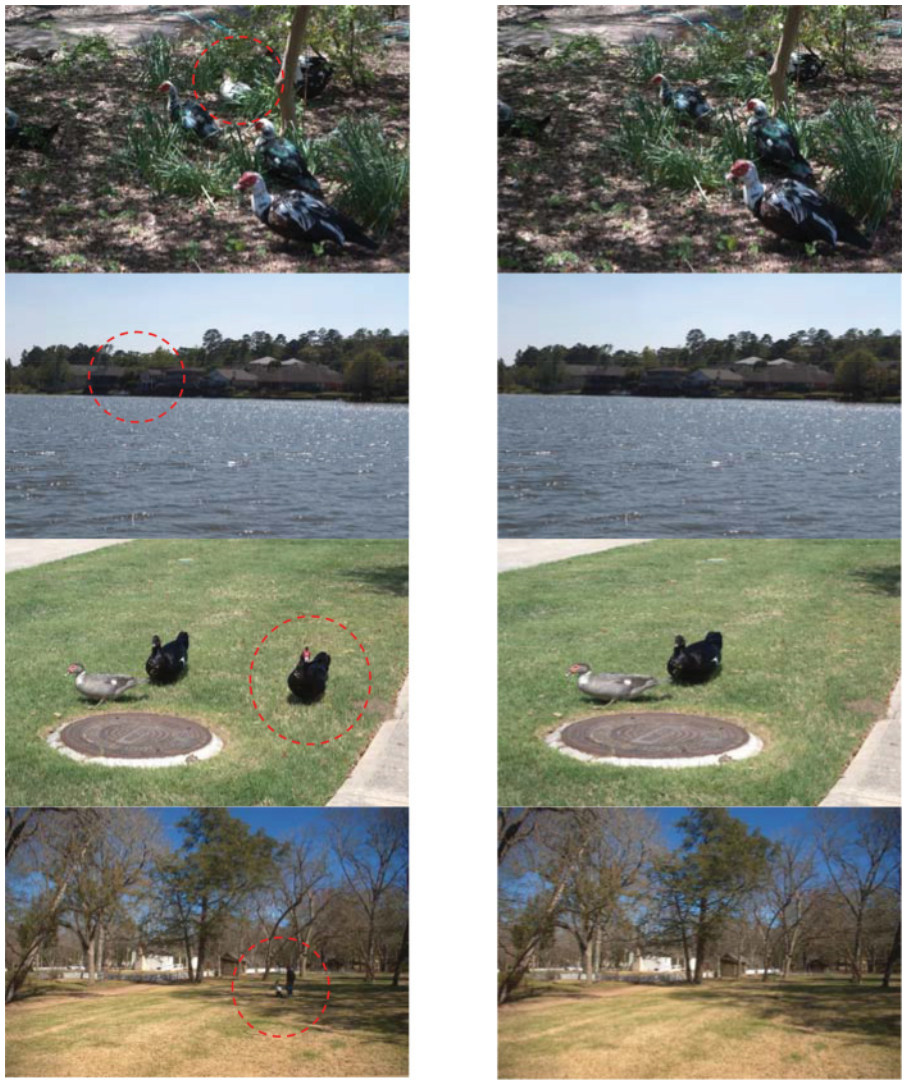

Fig. 7. Image samples in our experiments. The untouched is shown on the left and the modified on the right. The objects highlighted by dashed circles on the left were removed by seam carving.

noticeably improves the detection performance. Overall, the rich-model-based detector is not accurate.

6.3. Detection of Seam-Carved Tampering in JPEG Images

6.3.1. Setup. We adopted 500 JPEG images with a standard quantization table of the quality "75." The seam carving forgery tool at the website http://code.google.com/p/seam-carving-gui/ is used to modify JPEG images. The small objects are removed from the images at first in the spatial domain by using the tool, and the doctored images are stored in JPEG at the same quality of the untouched image. This avoids double JPEG compression for possible exposure by the detection of double JPEG compression. Figure 7 shows several untouched images (on the left) and tampered images (on the right) by seam carving in our experiment.

TABLE IVA. Seam-Carved Forgery Detection
Mean testing accuracy (%) over 2,000 experiments with Fisher linear discriminant (**F**) and over 1,000 experiments with ensemble classifier (**E**).

	Mean Detection Accuracy, %	
Detector	F	E
CC-absNJ	94.8	94.8
absNJ	87.3	86.9
CC-PEV	85.6	92.7
PEV	87.7	87.8
Markov	88.2	92.0
CC-C300	93.7	90.8
CF	94.2	95.3
CC-JRM	95.9	96.5
SRMQ1	97.0	97.5
CC-JRM+SRMQ1	96.8	97.1
CC-absNJ+SRMQ1	**97.2**	**97.6**

While data embedding in JPEG-based steganography directly modifies quantized DCT coefficients in the transform domain, as we analyzed in Section 5, seam carving inserts or removes seams with minimum cost from one end of the image and modifies the pixel values directly in the spatial domain. The modification generally destructs the original JPEG compression block, resulting in the change of the joint density in the DCT domain. In addition to the approach of calibrated neighboring joint density features in the DCT domain, we make use of SRMQ1, a detector of spatial domain rich models originally designed to detect spatial domain-based steganography [Fridrich and Kodovsky 2012]. We surmise that SRMQ1 may capture the statistical modification in the spatial domain that was caused by seam carving; therefore, we integrate CC-absNJ with SRMQ1 to detect seam-carved tampering in JPEG images. Meanwhile, we conjecture that most steganalysis detectors are also effective in detecting this manipulation.

Because for this experiment our forgery database is relatively small, we significantly increase the number of experiments for classification. We perform the experiment for each detector 2,000 times with Fisher linear discriminant and 1,000 times with ensemble classifier. Generally, the computational cost by applying ensemble classifier to the detectors of rich models is much higher than Fisher linear discriminant. In each case, 50% untouched images and 50% doctored images are randomly selected for training, and the remainder is used for testing.

6.3.2. Results. The mean testing accuracy values are given in Table IVA with 10 combinations of different detectors. While all these detectors were originally designed to detect JPEG-based steganography, all are effective in discriminating seam-carved tampering from untouched images. The union of calibrated neighboring joint density CC-absNJ with the detector of spatial domain rich model SRMQ1 obtains the best detection accuracy.

6.4. Discussions

Compared to the calibration that only takes one-time cropping (e.g., only shifting by four rows and four columns), the computation cost of our proposed 63-time-cropping-based calibration is relatively high but obtains a better detection accuracy. Figure 8 compares the mean detection accuracy over 100 experiments by using neighboring joint density-based approaches with 63-time cropping with one-time cropping (cropping by four rows and four columns) and without any cropping. Compared to the original

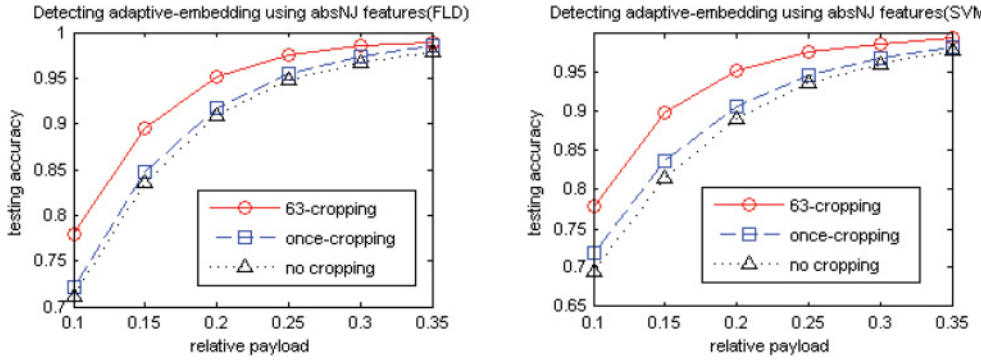

Fig. 8. Mean detection accuracy over 100 experiments using absNJ-based approaches with 63-time cropping, with one-time cropping, and without cropping.

neighboring joint density-based approach, the approach with one-time cropping improves the detection accuracy slightly, and the approach with 63-time cropping noticeably improves the detection performance.

It is worth noting that the 63-time-cropping–based approach is useful to generate the reference features for steganalysis and is also very promising in detecting misaligned cropping and recompression with the same quantization matrix and relevant forgery, including copy-paste and composite forgery, that are derived from the same camera source and encoded with the same quantization table [Liu 2011b].

In steganalysis of YASS, although Li et al. [2009] demonstrated the weakness of the YASS steganographic system, the detection algorithm does not search all candidate host blocks, resulting in deteriorated detection performance when detecting the steganograms produced by a large B-block parameter. Additionally, the detection assumes the condition of the exact positions of AC coefficients that are used for data embedding, which is generally inapplicable. Following the strategy to expose potential candidate blocks, our study has surmounted such obstacles by searching all possible candidate blocks and comparing the neighboring joint density of these candidate blocks and the noncandidate neighboring blocks.

In an original YASS embedding algorithm, the embedding is limited to the 19 low-frequency AC coefficients; the upper left of the first B-block is overlapped with the upper left of the first 8×8 block. If we assume that the embedding positions of binary hidden bits are not limited to the 19 low-frequency AC DCT coefficients, our approach is still effective for the detection because our feature extraction is not limited to the position of 19 low-frequency AC coefficients. However, if prior knowledge of approximate embedding position is available, the detection performance may be further improved, as shown in Table IIIE.

If the first B-block is randomly misplaced from the upper-left point of the first 8×8 block, we can exhaust all possibility of mismatching, a total of 64 combinations including the original exact matching; accordingly, we can retrieve the *diff-absNJ* features in each mismatching, which will detect such polymorphism of the YASS steganographic system. In this case, the detector will contain $64 \times 63 = 4{,}032$ features. However, the detector cannot deal with the completely randomized embedding if we further revise and improve the YASS algorithm.

A rich-model–based detector can be applied to detect YASS steganograms without exposing the position of candidate blocks, although the detection performance is not as accurate as our approach, and the computational cost is also fairly high with an ensemble classifier and too high to be suitable with SVM. However, a rich-model-based

approach demonstrates a direction to deal with completely randomized embedding that may be further investigated. Meanwhile, YASS detection is still difficult when the *noused* parameter is small.

To reduce the feature dimensionality and to further improve the detection accuracy, we may integrate all detectors together; a feature selection algorithm is applied to select the optimal feature set. The feature selection to reduce feature dimensionality and improve detection accuracy in steganalysis has been studied in our previous research [Liu et al. 2008a, 2010]. We do not apply any feature selection algorithm in this study to compare the detection performance under different combinations of features. There are many algorithms to select an optimal feature set and achieve the best classification performance, such as SVM-RFE [Guyon et al. 2002], MSVM-RFE [Zhou and Tuck 2007], recursive feature addition based on supervised learning and similarity measurement [Liu et al. 2009c], minimum Redundancy Maximum Relevance [Peng et al. 2005], and unified probabilistic model-based global and local unsupervised feature selection [Guan et al. 2011]. The steganalysis performance could be further improved by employing feature selection algorithms while obtaining an optimal feature set with reduced feature dimensionality, which could be applied to rich models.

7. CONCLUSIONS

Steganalysis and forgery detection in image forensics are commonly treated separately; however, in this research, we design a method targeting the detection of steganography and seam-carved forgery together in JPEG images. We analyze the neighboring joint density of the DCT coefficients and reveal the difference between an untouched image and the modified version. In real detection, an untouched image and the modified version may not be obtained at the same time, and different JPEG images may have different neighboring joint density features. To produce the self-calibration, we design the reference features of neighboring joint density features under different shift recompression and propose calibrated neighboring joint density-based approaches to distinguish steganograms and altered images from untouched ones. Our study shows that this approach has multiple promising applications in image forensics. Compared to the state of the art of steganalysis detectors, our approaches deliver better or comparable detection performances with a much smaller feature set to detect several steganographic systems, including DCT-embedding-based adaptive steganography and YASS. Our method is also effective in detecting seam-carved forgery in JPEG images. By integrating calibrated neighboring density with spatial domain rich models that were originally designed for steganalysis, the hybrid approach obtains the best detection accuracy to discriminate seam-carved forgery from an untouched image in JPEG format. Our study shows that it has a promising manner by exploring steganalysis and forgery detection together.

It is still hard to break YASS while the steganograms are produced by using a small *noused* parameter. In detecting seam-carving forgery, rich models provide a marked improvement with abundant features. In the future, we may introduce scale-invariant visual language modeling [Wu et al. 2009] and integrate the modeling with our proposed algorithms to further improve image tampering detection. The detection of completely randomized embedding in JPEG images will also be explored.

ACKNOWLEDGMENTS

The support for this study from the National Institute of Justice and National Science Foundation is highly appreciated. We thank anonymous reviewers for their insightful comments that enabled us to improve our work, and thank Dr. Bin Li for providing us the code for YASS feature extraction and Dr. Chunhua Chen for providing us the code for Markov feature extraction. Special thanks go to Dr. Jessica Fridrich and Dr. Jan Kodovsky for sharing their feature extraction and ensemble classifier via their website. We are also

grateful to Mr. Xiaodong Li for meticulously producing tampered JPEG images and to Ms. Delia Gallinaro and Ms. Ronda Harris of Sam Houston State University (SHSU) for their proofreading. The first author is also grateful to the SHSU Office of Research and Sponsored Programs for the support.

REFERENCES

S. Avidan and A. Shamir. 2007. Seam carving for content-aware image resizing. *ACM Transactions on Graphics* 26, 3, Article 10.

S. Bayram, A. E. Dirik, H. T. Sencar, and N. D. Memon. 2010. An ensemble of classifiers approach to steganalysis. In *Proceedings of International Conference on Pattern Recognition 2010*. 4376–4379.

BOSS. http://bows2.ec-lille.fr/.

C. C. Chang and C. J. Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems Technology*, 2, 3, Article 27.

I.-C. Chang, J.-W. Yu, and C.-C. Chang. 2013. A forgery detection algorithm for exemplar-based inpainting images using multi-region relation. *Image and Vision Computing* 31, 1, 57–71.

B. Chen and G. W. Wornell. 2001. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transaction on Information Theory* 47, 4, 1423–1443.

Y. Chen and C. Hsu. 2011. Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection. *IEEE Transactions on Information Forensics and Security* 6, 2, 396–406.

C. Chen and Y. Shi. 2008. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Proc. 2008 IEEE International Symposium on Circuits and Systems*, 3029–3032.

D. Choand and T. Bui. 2005. Multivariate statistical modeling for image denoising using wavelet transforms. *Signal Processing: Image Communication*, 20, 1 (Jan. 2005), 77–89.

digiKam.org. http://www.digikam.org/node/439.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9 (June 2008), 1871–1874.

T. Filler and J. Fridrich. 2010. Gibbs construction in steganography. *IEEE Transactions on Information Forensics and Security* 5, 4 (Sep. 2010), 705–720.

T. Filler and J. Fridrich. 2011. Design of adaptive steganographic schemes for digital images. In *Proceedings of SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics XIII*.

T. Filler, J. Judas, and J. Fridrich. 2011. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security* 6, 3 (Sep 2011), 920–935.

C. Fillion and G. Sharma. 2010. Detecting content adaptive scaling of images for forensics applications. *Proceedings of SPIE, Media Forensics and Security II*. 7541.

J. Fridrich. 2004. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In *Proceedings of the Information Hiding Workshop.* 67–81.

J. Fridrich, J. Kodovsky, V. Holub, and M. Goljan. 2011a. Breaking HUGO—the process discovery. In *Proceedings of the 13th Information Hiding Workshop.* 85–101.

J. Fridrich, J. Kodovsky, V. Holub, and M. Goljan. 2011b. Steganalysis of content-adaptive steganography in spatial domain. In *Proceedings of the 13th Information Hiding Workshop.* 102–117.

J. Fridrich and J. Kodovsky. 2012. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security* 7, 3, 868–882.

D. Fu, Y. Shi, D. Zou, and G. Xuan. 2006. JPEG steganalysis using empirical transition matrix in block DCT domain. In *Proceedings of the IEEE 8th Workshop on Multimedia Signal Processing*. 310–313.

Y. Guan, J. G. Dy, and M. I. Jordan. 2011. A unified probabilistic model for global and local unsupervised feature selection. In *Proceedings of the 28th International Conference on Machine Learning.* 1073–1080.

G. Gul and F. Kurugollu. 2011. A new methodology in steganalysis: Breaking highly undetectable steganography (HUGO). In *Proceedings of the 13th Information Hiding Workshop.* 71–84.

I. Guyon, J. Weston, S. Barnhill, and V. N. Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 1–3, 389–422.

F. Heijden, R. P. W. Duin, D. Ridder, and D. M. J. Tax. 2004. *Classification, Parameter Estimation and State Estimation-An Engineering Approach Using Matlab*. John Wiley & Sons.

S. Hetzl and P. Mutzel. 2005. A graph-theoretic approach to steganography. In *Proceedings of the 9th IFIP TC-6 TC-11 International Conference on Communication and Multimedia Security*. 119–128.

J. M. Hilbe. 2009. *Logistic Regression Models.* Chapman & Hall/CRC Press.

imagemagick.org. Available at http://www.imagemagick.org/Usage/resize/#liquid-rescale.

Iresizer.com. Available at http://www.iresizer.com/.

T. Joachims. 2002. Estimating the generalization performance of a SVM efficiently. In *Proceedings of the 17th International Conference on Machine Learning*. 431–438.

Justice.gov. 2010a. Available at http://www.justice.gov/opa/documents/062810complaint1.pdf.

Justice.gov. 2010b. Available at http://www.justice.gov/opa/documents/062810complaint2.pdf.

A. Ker. 2004. Improved detection of LSB steganography in grayscale images. In *Proceedings of the 6th Information Hiding Workshop*. 97–115.

M. Kharrazi, H. T. Sencar, and N. D. Memon. 2006. Improving steganalysis by fusion techniques: A case study with image steganography. *LNCS Transactions on Data Hiding and Multimedia Security* I, 123–137.

J. Kodovsky and J. Fridrich. 2009. Calibration revisited. In *Proceedings of the 11th ACM Multimedia and Security Workshop*. 63–74.

J. Kodovsky, T. Pevny, and J. Fridrich. 2010. Modern steganalysis can detect YASS. In *Proceedings of the SPIE, Electronic Imaging, Media Forensics and Security XII*. 02-01–02-11.

J. Kodovsky and J. Fridrich. 2011. Steganalysis in high dimensions: Fusing classifiers built on random subspaces. *Proceedings of SPIE* 7880, 78800L, 2011.

J. Kodovsky, J. Fridrich, and V. Holub. 2012. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security* 7, 2, 432–444.

J. Kodovsky and J. Fridrich. 2012. Steganalysis of JPEG images using rich models. *Proceedings of SPIE 8303*, Media Watermarking, Security, and Forensics 2012, 83030A (Feb. 9, 2012), doi:10.1117/12.907495

B. Li, Y. Shi, and J. Huang. 2009. Steganalysis of YASS. *IEEE Transactions on Information Forensics and Security*, 4, 3, 369–382, 2009.

liquidrescale.wikidot.com. http://liquidrescale.wikidot.com/en:examples.

Q. Liu, A. H. Sung, and B. Ribeiro. 2005. Statistical correlations and machine learning for steganalysis. *Proceedings of the 7th Adaptive and Natural Computing Algorithms*. 437–440, Springer, New York.

Q. Liu, A. H. Sung, J. Xu, and B. Ribeiro. 2006a. Image complexity and feature extraction for steganalysis of LSB matching steganography. In *Proceedings of the 18th International Conference on Pattern Recognition*. 2, 267–270.

Q. Liu, A. H. Sung, and M. Qiao. 2006b. Video steganalysis based on the expanded Markov and joint distribution on the transform domains—Detecting MSU StegoVideo. In *Proceedings of the 7th International Conference on Machine Learning and Applications*. 671–674.

Q. Liu and A. H. Sung. 2007. Feature mining and neuro-fuzzy inference system for steganalysis of LSB matching steganography in grayscale images. *Proc. 20th International Joint Conference on Artificial Intelligence*, 2808–2813, Jan. 2007.

Q. Liu, A. H. Sung, Z. Chen, and J. Xu. 2008a. Feature mining and pattern classification for steganalysis of LSB matching steganography in grayscale images. *Pattern Recognition* 41, 1, 56–66, 2008.

Q. Liu, A. H. Sung, B. Ribeiro, M. Wei, Z. Chen, and J. Xu. 2008b. Image complexity and feature mining for steganalysis of least significant bit matching steganography. *Information Sciences* 178, 1, 21–36, 2008.

Q. Liu, A. H. Sung, and M. Qiao. 2009a. Improved detection and evaluation for JPEG steganalysis. In *Proceedings of the 17th ACM Multimedia*. 873–876.

Q. Liu, A. H. Sung, and M. Qiao. 2009b. Novel stream mining for audio steganalysis. In *Proceedings of the 17th ACM Multimedia*. 95–104.

Q. Liu, A. H. Sung, Z. Chen, J. Liu, X. Huang, and Y. Deng. 2009c. Feature selection and classification of MAQC-II breast cancer and multiple myeloma microarray gene expression data. *PLoS ONE* 4, 12, e8250.

Q. Liu, A. H. Sung, M. Qiao, Z. Chen, and B. Ribeiro. 2010. An improved approach to steganalysis of JPEG images. *Information Sciences* 180, 9, 1643–1655.

Q. Liu, A. H. Sung, and M. Qiao. 2011a. Neighboring joint density-based JPEG steganalysis. *ACM Transactions on Intelligent Systems Technology* 2, 2, Article 16.

Q. Liu, A. H. Sung, and M. Qiao. 2011b. Derivative-based audio steganalysis. *ACM Transactions on Multimedia Computing, Communications, and Applications* 7, 3, Article 18.

Q. Liu, A. H. Sung, and M. Qiao. 2011c. A method to detect JPEG-based double compression. In *Proceedings of the 8th International Conference on Advances in Neural Networks*, II. 466–476.

Q. Liu. 2011a. Steganalysis of DCT-embedding based adaptive steganography and YASS. In *Proceedings 13th ACM Multimedia Workshop on Multimedia and Security*. 77–86.

Q. Liu. 2011b. Detection of misaligned cropping and recompression with the same quantization matrix and relevant forgery. *Proc. 3rd ACM Workshop on Multimedia in Forensics and Intelligence*, 25–30, 2011.

Q. Liu, P. A. Cooper, L. Chen, H. Cho, Z. Chen, M. Qiao, Y. Su, M. Wei, and A. H. Sung. 2013. Detection of JPEG double compression and identification of smartphone image source and post-capture manipulation. *Applied Intelligence* 39, 4, 705–726.

S. Lyu and H. Farid. 2005. How realistic is photorealistic. *IEEE Transactions on Signal Processing* 53, 2, 845–850.

L. Marvel, C. Boncelet, and C. Retter. 2002. Spread spectrum image steganography. *IEEE Transactions on Image Processing* 8, 8, 1075–1083.

J. Mielikainen. 2006. LSB matching revisited. *IEEE Signal Processing Letters* 13, 5 (May 2006), 285–287.

X. Pan and S. Lyu. 2010. Region duplication detection using image feature matching. *IEEE Transactions on Information Forensics and Security* 5, 4, 857–867.

X. Pan, X. Zhang, and S. Lyu. 2012. Exposing image splicing with inconsistent local noise variances. In *Proceedings of the 2012 IEEE International Conference on Computational Photography*. 1–10.

H. Peng, F. Long, and C. Ding. 2005. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 8 (Aug. 2005), 1226–1238.

T. Pevny and J. Fridrich. 2007. Merging Markov and DCT features for multi-class JPEG steganalysis. *Proceedings of SPIE* 6505, 650503, 2007; doi:10.1117/12.696774

T. Pevny and J. Fridrich. 2008. Detection of double-compression in JPEG images for applications in steganography. *IEEE Transactions on Information Forensics and Security* 3, 2, 247–258.

T. Pevny, T. Filler, and P. Bas. 2010. Using high-dimensional image models to perform highly undetectable steganography. In *Proceedings of the 12th Information Hiding Workshop*. 161–177.

Photoshopsupport.com. Available at http://www.photoshopsupport.com/photoshop-cs4/what-is-new-in-photoshop-cs4.html.

N. Provos. 2001. Defending against statistical steganalysis. In *Proceedings of the 10th USENIX Security Symposium*. 10, 323–335.

M. Qiao, A. H. Sung, and Q. Liu. 2013. MP3 audio steganalysis. *Information Sciences* 231, 123–134.

V. Sachnev, H. J. Kim, and R. Zhang. 2009. Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome code. In *Proceedings of the 11th ACM Multimedia & Security Workshop*. 131–140.

P. Sallee. 2003. Model-based steganography. In *Proceedings of the 2003 International Workshop on Digital Watermarking*. 154–167.

P. Sallee. 2005. Model-based methods for steganography and steganalysis. *International Journal of Image and Graphics* 5, 1, 167–189.

A. Sarkar, L. Nataraj, and B. Manjunath. 2009. Detection of seam carving and localization of seam insertions in digital images. *Proc. 11th ACM Multimedia and Security Workshop*, 107–116, 2009.

A. Sarkar, K. Solanki, and B. Manjunath. 2010. Obtaining higher rates for steganographic schemes while maintaining the same detectability. *Proc. 12th International Conference on Information Hiding*, 178–192, 2010.

Y. Shi, C. Chen, and W. Chen. 2006. A Markov process based approach to effective attacking JPEG steganography. *Proc. 8th Information Hiding Workshop*, 249–264, 2006.

K. Solanki, A. Sarkar, and B. Manjunath. 2007. YASS: Yet another steganographic scheme that resists blind steganalysis. *Proc. 9th Information Hiding Workshop*, 16–31, 2007.

V. Vapnik. 1998. *Statistical Learning Theory*. John Wiley, 1998.

A. Westfeld. 2001. High capacity despite better steganalysis (F5 – a steganographic algorithm). *Proc. 4th Information Hiding Workshop*, 289–302, 2001.

L. Wu, Y. Hu, M. Li, N. Yu, and X.-S. Hua. 2009. Scale-invariant visual language modeling for object categorization. *IEEE Transaction on Multimedia*, 11, 2, 286–294.

R. Yan. 2006. *MATLABArsenal, a MATLAB Package for Classification Algorithms*. Available at http://www.4shared.com/get/pYaCFOch/MATLABArsenal.html.

X. Zhou and D. P. Tuck. 2007. MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data. *Bioinformatics* 23, 9, 1106–1114.

Received November 2012; revised May 2013; accepted June 2013