Open access • Proceedings Article • DOI:10.1117/12.759155

Estimation of primary quantization matrix for steganalysis of double-compressed JPEG images — Source link

Tomáš Pevný, Jessica Fridrich

Institutions: Binghamton University

Published on: 14 Feb 2008 - electronic imaging (International Society for Optics and Photonics)

Topics: Quantization (image processing), Steganalysis, Lossless JPEG, JPEG and Discrete cosine transform

Estimation of Primary Quantization Matrix for Steganalysis of Double-Compressed JPEG Images

Tomáš Pevný[a], Jessica Fridrich[b]

[a]Department of Computer Science, Binghamton University, State University of New York
[b]Department of Electrical and Computer Engineering, Binghamton University, State University of New York *

ABSTRACT

A JPEG image is double-compressed if it underwent JPEG compression twice, each time with a different quantization matrix but with the same 8×8 grid. Some popular steganographic algorithms (Jsteg, F5, OutGuess) naturally produce such double-compressed stego images. Because double-compression may significantly change the statistics of DCT coefficients, it negatively influences the accuracy of some steganalysis methods developed under the assumption that the stego image was only single-compressed. This paper presents methods for detection of double-compression in JPEGs and for estimation of the primary quantization matrix, which is lost during recompression. The proposed methods are essential for construction of accurate targeted and blind steganalysis methods for JPEG images, especially those based on calibration. Both methods rely on support vector machine classifiers with feature vectors formed by histograms of low-frequency DCT coefficients.

1. MOTIVATION

In this paper, we consider a JPEG image double-compressed if it was compressed twice, each time with a different quantization matrix. The quantization matrix used in the first compression is called the primary quantization matrix, the quantization matrix used in subsequent (second) compression is called the secondary quantization matrix. Since the JPEG image file does not keep information about the compression history, only the latest (secondary) quantization matrix is stored within the file and the primary quantization matrix is lost.

Detection of double-compression is important in steganalysis as well as in forensics because the fact that an image was double-compressed indicates that it was manipulated. By determining double-compression history in smaller regions, we may discover traces of malicious manipulation. For example, when pasting an object into a decompressed JPEG and resaving with a different JPEG quality factor, the pasted object may exhibit different repetitive JPEG compression artifacts than the rest of the image.

Some steganographic algorithms (e.g., F5[21] and OutGuess[18]) decompress the cover image to the spatial domain and then the image is compressed again during embedding with a user supplied or a default quality factor. Unless the quantization matrices match, the resulting stego image will be double-compressed. Thus, steganalytic methods also benefit from knowledge of stego image compression history. This is especially true for methods that use calibration[5] to estimate the statistics of the cover image. It is absolutely essential to adjust the calibration to mimic what happened during embedding. To do so, we need to accurately detect double-compressed images and estimate their primary quantization matrix, otherwise the steganalytic methods may give completely misleading results.[5]

In this paper, we address two problems: the detection of double-compression in JPEG images and the estimation of primary quantization matrix. Even though the first problem can be understood as a subproblem of the second one, we consider them separately. This allows us to achieve more accurate double-compressed detection.

This paper is organized as follows. In Section 2, we review the basics of JPEG compression relevant for this paper and define double-compression together with its effects on DCT coefficients. Previous approaches to the problem of double-compression detection and primary quantization steps estimation are described in Section 3. Section 4 presents our solutions to both problems. Experimental results are described in Section 5. Finally, Section 6 concludes the paper.

*Jessica Fridrich: E-mail: fridrich@binghamton.edu, Telephone: +1 607 777 6177, Fax: +1 607 777 4464

2. EFFECTS OF DOUBLE-COMPRESSION

Here, we introduce notation used in this paper by reviewing certain facts about JPEG compression[12] that are relevant for recovering the compression history of JPEG images. Then, we describe effects of double-compression on quantized DCT coefficients.

2.1. Basics of JPEG compression

JPEG compression starts by grouping pixels into disjoint 8×8 blocks $B_{rs},\ r, s = 0, \ldots, 7$, followed by the Discrete Cosine Transformation (DCT) applied independently to each block B

$$d_{ij} = \sum_{r,s=0}^{7} \frac{w(r)w(s)}{4} \cos\frac{\pi}{16} r(2i+1) \cos\frac{\pi}{16} s(2j+1) B_{rs},$$

where $w(0) = \frac{1}{\sqrt{2}}$, and $w(r > 0) = 1$. The next step implements the lossy part of JPEG compression, where coefficients d_{ij} are divided by quantization steps from the quantization matrix Q_{ij} and rounded to integers

$$D_{ij} = \left[\frac{d_{ij}}{Q_{ij}}\right],\ i, j \in \{0, \ldots, 7\}, \tag{1}$$

where $[\cdot]$ denotes rounding. We denote the i, j-th DCT coefficient in the k-th block as $D^k_{ij},\ k \in \{1, \ldots .l\}$, where l is the number of all 8×8 blocks in the image. The pair $(i, j) \in \{0, \ldots, 7\} \times \{0, \ldots, 7\}$ is called the *spatial frequency* (or *mode*) of the DCT coefficient. The quantized coefficients D^k_{ij} in each 8×8 block are ordered along the zig-zag path, encoded, and losslessly compressed.

Decompression works in the reverse order. Blocks of quantized DCT coefficients are recovered from the JPEG file and then multiplied by quantization steps, $\hat{d}_{ij} = Q_{ij} \cdot D_{ij}$. The resulting values are transformed using the Inverse Discrete Cosine Transformation (IDCT) and rounded to integers in the range $[0, 255]$ (for 8 bit images). The decompressed block $\hat{B}$ is

$$\hat{B}_{ij} = \operatorname{trunc}\left([\mathrm{IDCT}(Q_{ij} \cdot D_{ij})]\right),\ i, j \in \{0, \ldots, 7\}. \tag{2}$$

The reference implementation of JPEG compression provided by Independent JPEG Group[†] recommends a set of matrices indexed by a *quality factor* from the set $\{1, 2, \ldots, 100\}$. We refer to these matrices as *standard* matrices.

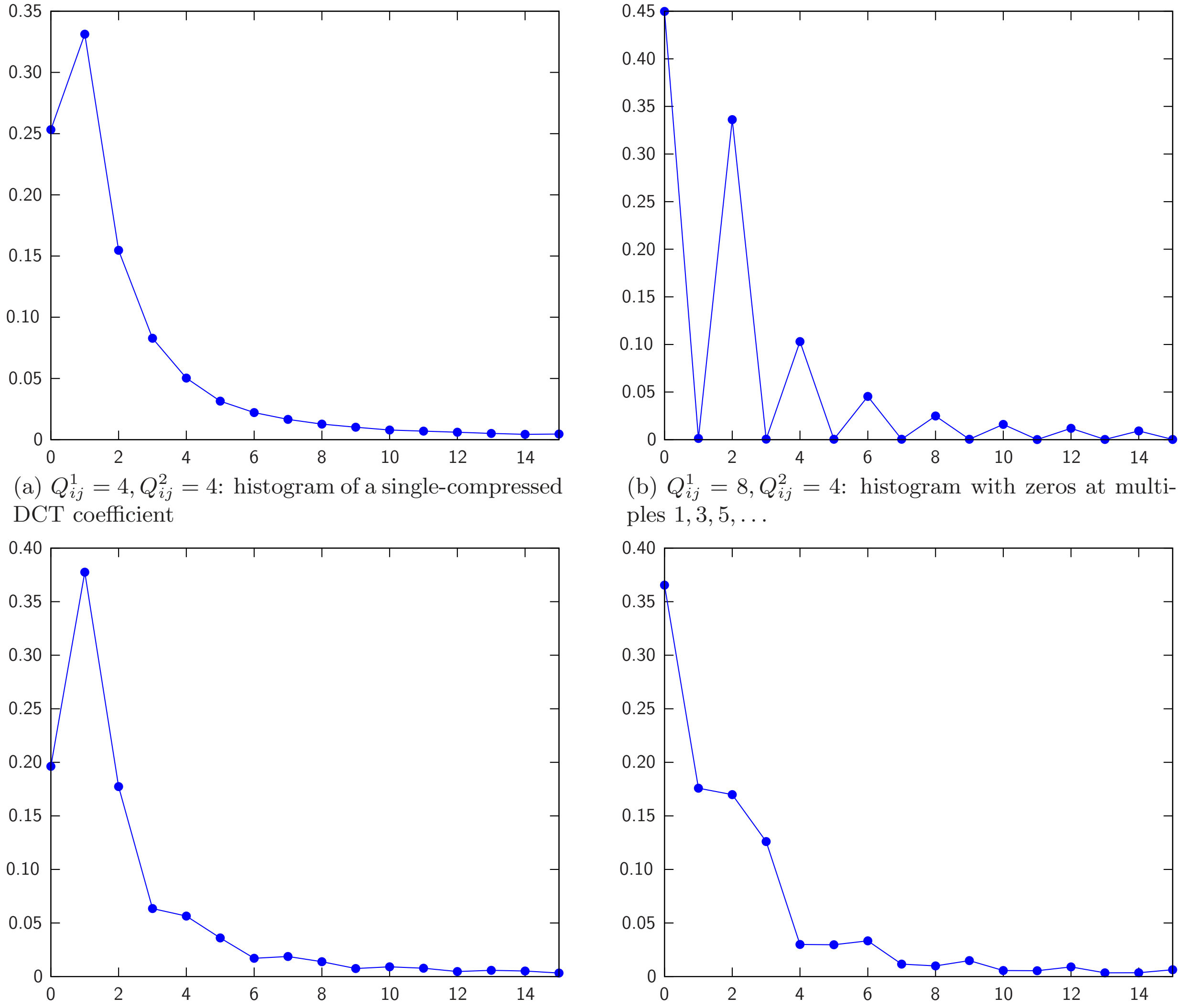
2.2. Effect of double-compression on DCT histograms

We call a JPEG image *double-compressed* if it underwent JPEG compression twice with the same alignment of 8×8 blocks, but each time with a different quantization matrix. This definition is motivated by the embedding mechanism of steganographic techniques that produce such images. Technically, an image compressed twice, each time with a different alignment of the blocks is also "double-compressed", but it would not be recognized as such according to the definition in this paper. The quantization matrix Q^1 used during the first compression is called the *primary quantization matrix*. The quantization matrix Q^2 used in subsequent JPEG compression is called the *secondary quantization matrix*. Additionally, we say that a specific DCT *coefficient* D_{ij} was double-compressed if and only if $Q^1_{ij} \neq Q^2_{ij}$. Omitting the truncation in (2), the double compressed DCT coefficient D_{ij} is

$$D_{ij} = \left[\left[\frac{d_{ij}}{Q^1_{ij}}\right] \cdot \frac{Q^1_{ij}}{Q^2_{ij}}\right]. \tag{3}$$

From (3), we can see that the values of double-compressed DCT coefficients depend on the combination of quantization steps Q^1_{ij} and Q^2_{ij}. The effects of double-compression on the DCT coefficients D_{ij} for a fixed mode i, j are easily apparent in histograms

[†] `ftp://ftp.simtel.net/pub/simtelnet/msdos/graphics/jpegsr6.zip`



(a) $Q^1_{ij} = 4, Q^2_{ij} = 4$: histogram of a single-compressed DCT coefficient

(b) $Q^1_{ij} = 8, Q^2_{ij} = 4$: histogram with zeros at multiples $1, 3, 5, \dots$

(c) $Q^1_{ij} = 3, Q^2_{ij} = 4$: histogram with double peaks at multiples $(1, 2), (4, 5), (7, 8) \dots$

(d) $Q^1_{ij} = 6, Q^2_{ij} = 4$: histogram with double peaks at multiples $(1, 2), (4, 5), (7, 8), \dots$

Figure 1. Effect of double-compression on histograms of absolute values of DCT coefficients. The seconadary quantization coefficient is in all four cases the same $Q^2_{ij} = 4$, only the primary quantization coefficient Q^1_{ij} varies.

$$h_{ij}(m) = \sum_{k=0}^{l} \delta\left(\left|D^k_{ij}\right| - m \cdot Q^2_{ij}\right), \tag{4}$$

where $k \in \{1, \dots. l\}$ indexes the 8×8 block and δ is the indicator function, $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ when $x \neq 0$. We recognize two distinct artifacts of double-compression on the shape of the histogram $h_{ij}(m)$.

A *zero* occurs if there exists an integer $u > 0$ so that $Q^1_{ij} = u \cdot Q^2_{ij}$ (the primary quantization is coarser than the secondary quantization). In this case, $D_{ij} \approx u \cdot \left[\frac{d_{ij}}{Q^1_{ij}}\right]$ gives a restriction on the values of D_{ij}, which can now only attain values from the set $\{0, u, 2u, 3u, \dots\}$. Thus, $h_{ij}(m) = 0$ for $m \in \mathbb{N} \setminus \{0, u, 2u, 3u, \dots\}$. Formally,

$$h_{ij}(m) \begin{cases} > 0 & m \in \{0, k, 2k, 3k, \dots\} \\ = 0 & m \bmod u \neq 0. \end{cases}$$

Figure 1(b) shows an example of a histogram exhibiting zeros at $m \in \{1, 3, 5, 7, \ldots\}$ ($Q^1_{ij} = 8$ and $Q^2_{ij} = 4$).

A *double peak* occurs when there exist integers $u, v \geq 0$ so that $uQ^1_{ij} = \frac{1}{2}\left((v-1)Q^2_{ij} + vQ^2_{ij}\right)$ (the multiple $\left[\frac{d_{ij}}{Q^1_{ij}}\right] \cdot Q^1_{ij}$ falls in the middle of two multiples of Q^2_{ij} and no other multiple of Q^2_{ij} is closer). In this case, the multiple $\left[\frac{d_{ij}}{Q^1_{ij}}\right] \cdot Q^1_{ij}$ contributes approximately the same to both $(v-1)Q^2_{ij}$ and vQ^2_{ij}. Figures 1(c,d) show examples of double-peaks occurring at multiples $v = 2, 5, 8, \ldots$. A more detailed description of the impact of double-compression on the DCT histogram can be found in.[6, 16, 17]

We now point out an important case called *divisor*s, where there is no effect of double-compression of the value of the DCT coefficient despite the fact that the quantization steps are different. If there exists an integer $u > 0$ so that $Q^2_{ij} = u \cdot Q^1_{ij}$ (the complementary case to zeros, when the primary quantization is finer than the secondary quantization), then the shape of the histogram is almost identical to the histogram of single-compressed DCT coefficients quantized by Q^2_{ij}. In this case, the quantization step Q^1_{ij} cannot be detected and the DCT coefficient is not technically double-compressed.

While in a single-compressed JPEG image the histogram $h_{ij}(m)$ (Figure 1(a)) is well-modeled with a generalized Gaussian distribution,[11] histograms of double-compressed JPEG images do not follow this distribution. Since different primary quantization steps affect the shape of $h_{ij}(m)$ differently, it is possible to detect them.

3. PRIOR ART

The first publication on detection of double-compression is[6‡]. The authors recognized that it is impossible to restore all quantization steps Q^1_{ij}, since most DCT coefficients at higher frequencies are zeros and the available statistics is not sufficient for a reliable estimate. Instead, they focused on low frequencies $(i, j) \in \{(0, 1), (1, 1), (1, 0)\}$ and proposed three approaches. Two were based on the idea of matching the histograms $h_{ij}(m)$ to histograms obtained by simulated double-compression, where the raw image was estimated by calibration.[4, 5] The third approach utilized a collection of neural networks to detect patterns caused by different combinations of quantization steps Q^1_{ij} and Q^2_{ij} (Figure 1). The collection of neural networks consisted of individual networks constructed for each value of the secondary quantization step (SQS) of interest, $Q^2_{ij} \in \{1, \ldots, 9\}$ and detected the primary quantization steps (PQS) Q^1_{ij} in the range $[2, 9]$, for $Q^2_{ij} \in \{2, \ldots, 9\}$, and in the range $[1, 9]$ for $Q^2_{ij} = 1$. All neural networks used the same input feature vector

$$x = \{h_{ij}(2), h_{ij}(3), \ldots, h_{ij}(15)\}. \tag{5}$$

Interestingly, the authors did not use the histogram values for $m = \{0, 1\}$ in the feature vector. By empirical comparisons, the neural network approach performed the best. The reported accuracy on cover JPEG images was better than 99% for estimation of low frequency quantization steps with frequencies $(i, j) \in \{(0, 1), (1, 1), (1, 0)\}$, and better than 95% for quantization steps with frequencies $(i, j) \in \{(2, 0), (2, 1), (1, 2), (0, 2)\}$.

A different approach proposed in[16, 17] focused only on detection of double-compression instead of estimation of primary quantization steps. Similarly, the proposed solution was based on the histograms of DCT coefficients $h_{ij}(m)$. The authors showed that double-compression artifacts are periodic manifesting as peaks in the Fourier transform of $h_{ij}(m)$. The detection accuracy (double-compressed image detected as double-compressed) of their method was estimated on 100 cover images and was usually 100% with 0% false alarm rate (single-compressed image detected as double-compressed). Unfortunately, we were not able to obtain implementation of this approach in order to compare fairly it with our solution on the same database of images.

An interesting method for recovering the compression history of images was recently proposed in.[7] It is based on the claim that the distribution of the first digit of DCT coefficients in digital images of natural scenes follows the generalized Benford distribution

$$p(x) = N \cdot \log\left(1 + \frac{1}{1 + x^q}\right),$$

[‡] The problem of detection of previous (single) JPEG compression from bitmap images was also investigated in.[3]

where q is a free parameter and N is a normalization constant. After double-compression, the distribution of the first digit no longer follows the generalized Benford distribution. This fact was utilized to estimate the quantization matrix of (previously single-compressed) JPEG images available in some lossless image format, such as TIFF or PNG. The authors also suggested to form features from the histogram of the first digit of DCT coefficients (further called the Benford feature set) and train an SVM classifier to detect the primary quantization steps Q^1_{ij}. We investigate this approach in more detail in Section 5, where we compare it to the proposed method.

4. THE PROPOSED METHODS

All previously proposed methods targeted cover images only. For applications in steganalysis, however, it is important to recover compression history from stego images, whose statistics may be disturbed by embedding. The multi-classifier proposed in[14] consists of two separate classifiers and a double-compression detector serving as a pre-classifier. If the double-compression detector decides that an image has been double-compressed, it is sent to the multi-classifier targeted for double-compressed images that can only detect F5 and OutGuess and does not classify to other stego methods recognized by the multi-classifier for single-compressed JPEG images. Thus, the performance of the blind steganalyzer is greatly influenced by double-compression. The multi-classifier calls for a double-compression detector with a low probability of false positives, which means low probability of detecting a single-compressed image as double-compressed.

Double-compression detection can be understood as a sub-problem of the primary quantization step estimation. However, estimation of quantization steps is more difficult than detection of double-compression. To illustrate this, realize that the statistics available for double-compression detection can be extracted from all DCT coefficients, while estimating a primary quantization step can only use data from a given DCT mode.

Besides the double-compression detector, a good primary quality factor estimator (PQF) is also needed. Calibration of double-compressed images[14] must mimic the processing history of the image with the correct primary quantization matrix. The failure to accommodate the effects of double-compression may produce very inaccurate steganalysis results.[5]

The superior performance of the neural network approach reported in[6] steered our attention towards the use of tools of pattern recognition with features formed by histograms $h_{ij}(m)$. Because of the problem with insufficient statistics for high-frequency DCT coefficients, we limit the set of DCT frequencies used by both the DC detector and the PQF estimator to the set

$$\mathcal{L} = \{(1,0),(2,0),(3,0),(0,1),(1,1),(2,1),(0,2),(1,2),(0,3)\}. \tag{6}$$

Before continuing with detailed description of the proposed solutions, we note that by modeling the distribution of single-compressed DCT coefficients (e.g., using generalized Gaussian), from Eq. (3) it is certainly possible to derive a parametric distribution of double-compressed DCT coefficients for a fixed DCT mode. Thus, we could estimate Q^1_{ij} together with the nuisance parameters of the generalized Gaussian using maximum likelihood estimation. This approach, however, does not take into account the effects of embedding changes on the statistics of DCT coefficients, which can be quite complex and consequently hard to model (if possible at all in blind steganalysis setting). Since the stego images are our primer concern in this paper, we did not pursue this interesting idea further.

In the rest of this section, we first describe the double-compression detector and then the primary quantization matrix estimator.

4.1. Double-compression detector

The double-compression detector (DC detector) is an algorithm classifying images into two classes—single-compressed images and double-compressed images. We implemented it using a soft-margin Support Vector Machine[2] ($C-$SVM) with the Gaussian kernel $K(x,y) = \exp(-\gamma\|x-y\|^2)$. The feature vector is derived from histograms $h_{ij}(m), m \in \{0,\ldots,15\}$ (4) of absolute values of DCT coefficients D^k_{ij} in the inspected JPEG image. The histograms in the feature vector are calculated for the low-frequency DCT modes (6) because

SQS	Detectable PQS	#SVMs
4	$\mathcal{S}_4 = \{3, 4, 5, 6, 7, 8\}$	15
5	$\mathcal{S}_5 = \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$	36
6	$\mathcal{S}_6 = \{4, 5, 6, 7, 8, 9, 10, 11, 12\}$	36
7	$\mathcal{S}_7 = \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$	36
8	$\mathcal{S}_8 = \{3, 5, 6, 7, 8, 9, 10, 11, 12\}$	36

Table 1. Primary quantization steps (PQS) detectable by the multi-classifier for a given secondary quantization step (SQS). The last column (#SVMs) shows the number of binary Support Vector Machines in the multi-classifier.

the statistics at higher frequency modes is not sufficient for accurate detection for most secondary quality factors. The feature vector x can be formally written as

$$x = \left\{ \frac{1}{C_{ij}} \left(h_{ij}(0), h_{ij}(1), \ldots, h_{ij}(15)\right) \middle| \; (i,j) \in \mathcal{L} \right\},$$

where C_{ij} are normalization constants $\left(C_{ij} = \sum_{m=0}^{15} h_{ij}(m)\right)$. The dimension of the feature vector x is $16 \times |\mathcal{L}| = 144$.

Even though the proposed approach is applicable to JPEG images with any secondary quality factor, we only created the double-compression detector for two secondary quality factors 75 and 80 (the defaults of F5 and OutGuess) by training a special C-SVM for each secondary quality factor (SQF). We opted for this design with separate classifiers over the design with one large classifier, because it offers higher accuracy and faster training.

4.2. Primary quality factor estimator

Because we cannot estimate the whole primary quantization matrix for most quality factors due to insufficient data for high spatial frequencies in the JPEG file, we divided the primary quality factor estimator (PQF estimator) into two parts. The first part detects selected primary quantization steps for low-frequency AC modes $\mathcal{L}$, the second part finds the closest standard quantization matrix (as defined in Section 2) in the maximum likelihood sense.

4.2.1. Detector of primary quantization steps

The detector of the primary quantization steps is implemented as a collection of SVM-based multi-classifiers $\mathcal{F}_{Q^2_{ij}}$ indexed by the value of the secondary quantization step Q^2_{ij}. The standard quantization matrices for quality factors 75 and 80 only have 5 different quantization steps $\{4, 5, 6, 7, 8\}$ for modes $(i, j) \in \mathcal{L}$. Thus, we will need to construct 5 multi-classifiers. Each multi-classifier $\mathcal{F}_{Q^2_{ij}}$ classifies into $n_{Q^2_{ij}}$ classes, where $n_{Q^2_{ij}}$ is the number of all possible values of the primary quantization step when the secondary quantization step is Q^2_{ij}. Theoretically, $n_{Q^2_{ij}} = 255$ since the quantization step can be represented by an 8-bit number and it cannot be zero. In practice, we narrow the number of detected primary quantization steps to steps from 34 quantization matrices with quality factors from

$$\mathcal{Q}_{34} = \{63, 64, \ldots, 93, 94, 96, 98\}.$$

This set was determined by our database of available JPEG images. The quantization steps detected for each secondary quantization step are shown in Table 1. The number of SVMs is $\binom{n_{Q^2_{ij}}}{2}$ because we need one SVM for each pair of classes.

The feature vector x for the multi-classifier $\mathcal{F}_{Q^2_{ij}}$ is again formed by the histogram of absolute values of the first 16 multiples of Q^2_{ij} of all DCT coefficients $|D^k_{ij}|$ for all blocks $k = 1, \ldots, l$

$$x = \frac{1}{C}(h_{ij}(0), h_{ij}(1), \ldots, h_{ij}(15)), \tag{7}$$

where $C = \sum_{m=0}^{15} h_{ij}(m)$ is a normalization constant. The multi-classifier $\mathcal{F}_{Q^2_{ij}}$ is a collection of binary classifiers, where the outcome of the whole classifier is determined by voting (the "max-wins" scheme[10]). All binary classifiers are soft-margin Support Vector Machines (C-SVM) with Gaussian kernel.

Note that the feature vector cannot detect the cases when Q^1_{ij} is a divisor of Q^2_{ij}, when $Q^1_{ij} = 1$, or when $Q^1_{ij} = Q^2_{ij}$ because the normalized histograms are almost identical in this case. We classify all three cases into one common class $Q^1_{ij} = Q^2_{ij}$. Even though this is a fundamental limitation of the detector, it does not present problem for subsequent steganalysis, since the histograms of double-compressed DCT coefficients are almost the same.

4.2.2. Matching the closest standard quantization matrix

Denoting the estimated and the true primary quantization steps as $\hat{Q}^1_{ij}$ and Q^1_{ij}, respectively, the closest standard quantization matrix is obtained as

$$\hat{Q} \quad = \quad \arg\max_{Q \in \mathcal{T}} \prod_{i,j \in \mathcal{L}} P(\hat{Q}^1_{ij} | Q_{ij}, Q^2_{ij}),$$

where $\mathcal{T}$ is the set of all standard quantization matrices. Since the number of quality factors is finite and the calculation of the likelihoods is fast, we can find the maximum by exhaustive search over all $Q \in \mathcal{T}$. The conditional probabilities $P(\hat{Q}^1_{ij} | Q_{ij}, Q^2_{ij})$ are the probabilities that the classifier detects the primary quantization step $\hat{Q}^1_{ij}$ when the correct primary quantization step is Q_{ij} and the secondary quantization step is Q^2_{ij}. We estimate these probabilities empirically on images from the training set.

5. EXPERIMENTAL RESULTS

Before presenting the results of the experiments, we describe the training and testing set used throughout the experiment.

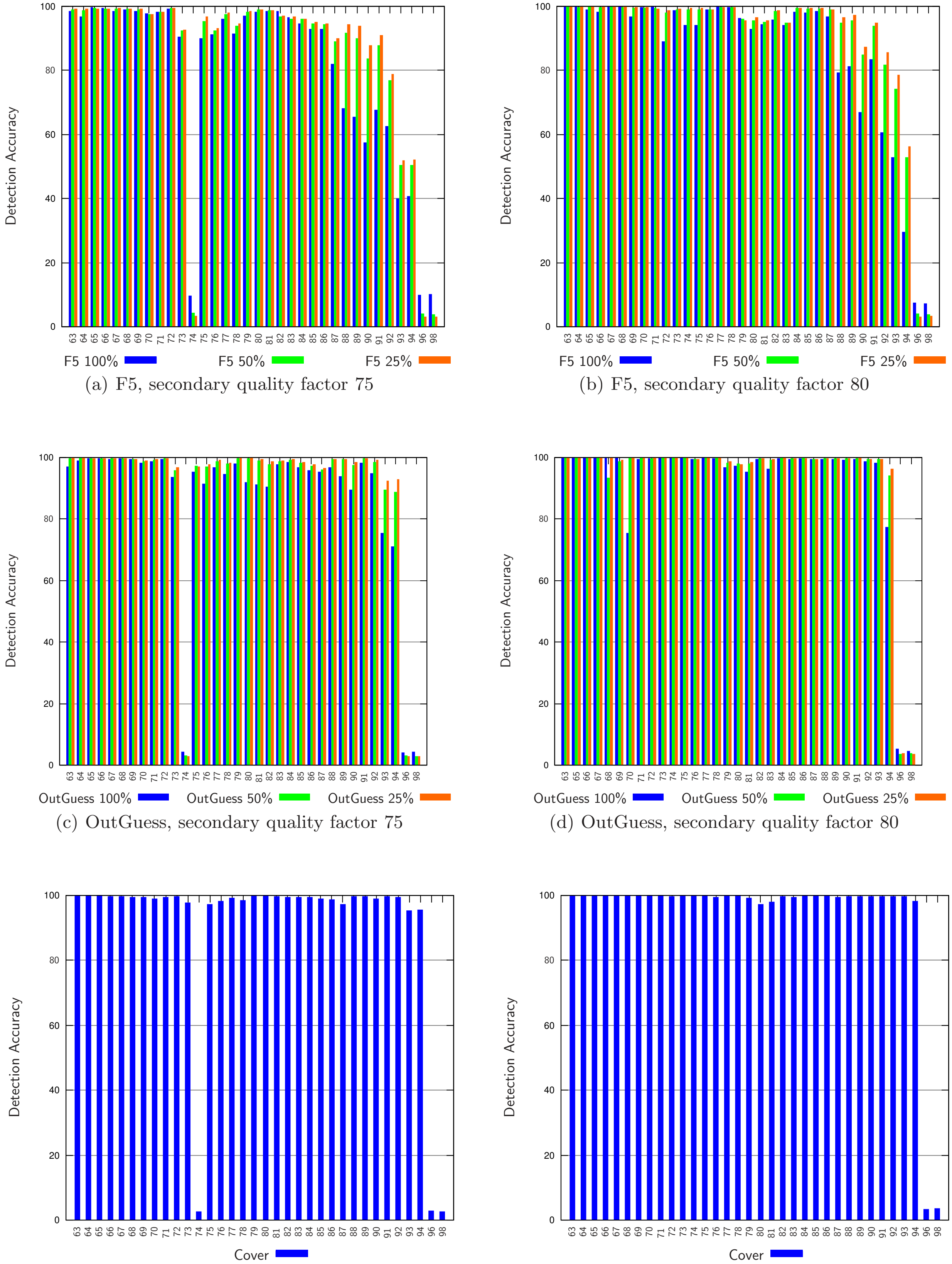
The training set was prepared from 3500 raw images. Double-compressed images were created by OutGuess and F5 with message lengths 100%, 50%, and 25% of their embedding capacity. We selected these two steganographic algorithms because their implementations produce double-compressed images *during embedding.* The double-compressed images had 34 different primary quality factors $\mathcal{Q}_{34} = \{63, 64, \ldots, 93, 94, 96, 98\}$ and two secondary quality factors 75 and 80, the defaults of OutGuess and F5. Single-compressed images with quality factors 75 and 80 were embedded by the following algorithms: F5, Model Based Steganography without[19] (MBS1) and with[20] deblocking (MBS2), JP Hide&Seek,[1] OutGuess, and Steghide.[8] The payloads were the same as for the double-compressed images except for MBS2, where the payload was 30% of the capacity of MBS1. Examples of cover images in the training set had the same combinations of quality factors.

The testing set was created in exactly the same manner from a disjoint set of 2506 images. Consequently, the total number of images with different compression history in the database was $|\mathcal{Q}_{34}| \times 2 \times 7 \times 6006 + 17 \times 6006 \approx 3,000,000$.

5.1. Double-compression detector

The classifiers in the double-compression detector were trained on 10000 examples of single-compressed and 10000 examples of double-compressed images. Images in the training set were selected randomly with uniform distribution of steganographic algorithms, message lengths, and primary quality factors in the case of double-compressed images. We did not use images whose histograms of modes from $\mathcal{L}$ did not exhibit double-compression artifacts (the cases when SQS were divisors of the PQS). These were images with SQF 75 and PQF $74, 75, 96, 98$ and images with SQF 80 and PQF $80, 96, 98$. The hyper-parameters of the C-SVMs (Gaussian kernel width γ and penalty C of incorrectly classified training samples) were determined by grid-search combined with 5-fold cross-validation[9] on the multiplicative grid

$$(C, \gamma) \in \left\{(2^i, 2^j) | i \in \{0, \ldots, 19\}, j \in \{-7, \ldots, 5\}\right\}.$$



(a) F5, secondary quality factor 75

(b) F5, secondary quality factor 80

(c) OutGuess, secondary quality factor 75

(d) OutGuess, secondary quality factor 80

(e) Cover, secondary quality factor 75

(f) Cover, secondary quality factor 80

Figure 2. Accuracy of the double-compression detector for secondary quality factors 75 and 80 on double-compressed cover images and images embedded with F5 and OutGuess algorithms. The x axis is the primary quality factor.

Quality factor	75	80
F5 100%	90.09%	92.87%
F5 50%	95.50%	95.67%
F5 25%	96.77%	96.69%
JP Hide&Seek 100%	95.99%	99.12%
JP Hide&Seek 50%	97.34%	98.82%
JP Hide&Seek 25%	97.37%	98.19%
MBS1 100%	93.25%	98.37%
MBS1 50%	96.43%	98.28%
MBS1 30%	96.73%	98.18%
MBS2 30%	97.05%	98.23%
OutGuess 100%	95.42%	97.29%
OutGuess 50%	97.25%	98.10%
OutGuess 25%	97.20%	97.84%
Steghide 100%	97.30%	98.19%
Steghide 50%	97.75%	97.95%
Steghide 25%	97.59%	97.94%
Cover	97.22%	97.35%

Table 2. Detection accuracy of double-compression detector on single-compressed JPEG images embedded by various steganographic algorithms with different payloads.

Figure 2 shows the accuracy of the double-compression detector on double-compressed images from the testing set (cover images and images embedded with F5 and OutGuess algorithms). The accuracy of the detector is generally above 95% and decreases for higher primary quality factors (92 and above). This decrease is to be expected because the double-compression artifacts in histograms of DCT coefficients are becoming more subtle. Moreover, increasingly more modes start sufferring from the "divisor" problem described in Section 4. Also notice that with higher embedding rate, the classification accuracy also decreases. This is especially true for images fully embedded with F5. This loss of accuracy is most likely due to the fact that F5 considerably changes the shape of the histogram because of the the shrinkage.

Notice that images with PQF 74 and SQF 75 were not recognized by the double-compression detector as double-compressed. This is because all quantization steps for modes from $\mathcal{L}$ are identical. A similar situation occurs for images with PQF 96 and 98 regarding their SQF. Since all quantization steps for modes from $\mathcal{L}$ are equal to 1, the histograms again do not exhibit traces of double-compression. Because the DCT coefficients are negligibly influenced by double-compression, we do not consider these two cases as a failure of the double-compression detector.

The overall accuracy on single-compressed JPEG images, shown in Table 2, is generally above 96% for all stego algorithms with the exception of fully embedded F5 images. The accuracy gap of about 6% on images fully embedded by F5 is due to the modification of the DCT histogram by embedding, as already discussed above. Notice that for images embedded with F5 with shorter message length, the detection accuracy is comparable to other algorithms. This improvement in accuracy is attributed to the decreased number of embedding changes due to matrix embedding.

We next compared the proposed approach (that we call Multiple-counting features) to the double-compression detector that uses the Benford feature set (Section 3) by training two separate C-SVM classifiers—one employing the multiple-counting features and the other employing Benford features. The training set of both classifiers contained 3400 examples of single-compressed and 3400 examples of double-compressed images, all with (secondary) quality factor 75. We excluded images with primary quality factor 74, 96, and 98 from training and testing sets because DCT coefficients with spatial frequencies in $\mathcal{L}$ are not technically double-compressed, as we explained above.

The classifiers were compared using the minimum total error under equal priors, $P_E = 0.5(P_{FA} + P_{MD})$, where P_{FA} is the probability of a single-compressed image being recognized as double-compressed and P_{MD} is the opposite. Errors were calculated over all images from our database with SQF 75, PQF 74, 96, and 98 excluded. The error, P_E, of the classifier employing multiple-counting features was 2.12% without any bias, while for the classifier employing Benford features $P_E = 53.67\%$ with a bias towards detecting images as double-compressed. The results indicate that on our database, Benford features cannot be used for detection of double-compression, as the performance of the classifier utilizing them is essentially equivalent to random guessing.

5.2. Primary quantization step detector

The binary $C-$SVMs in the multi-classifiers $\mathcal{F}_{Q^2_{ij}}$ for estimation of the primary quantization steps were trained on images classified by the DC-detector as double-compressed. The training set of each C-SVM contained 20000 examples—10000 examples from each class with a given combination of primary Q^1_{ij} and secondary Q^2_{ij} quantization steps. As in the case of the DC-detector, the hyper-parameters of C-SVMs were determined by grid-search over the multiplicative grid

$$(C, \gamma) \in \left\{(2^i, 2^j) | i \in \{4, \ldots, 18\}, i \in \{-8, \ldots, 6\}\right\},$$

where we evaluated each point by a 5-fold cross-validation. The search was performed independently for each C-SVM, which is rather costly, but it ensures the best results.

We compared the accuracy of the proposed primary quantization step detector to the Neural Network (NN) approach[6] on our database of testing images. The proposed detector performed better, especially on stego images. This is not surprising because our estimator was also trained on stego images and because SVM classifiers generally perform better than Neural Network classifiers (unless Neural Networks are carefully tuned for a given problem). The rare cases, where the Neural Network estimator performed better all corresponded to the divisor cases, which have negligible influence on steganalysis.

5.3. Primary quality factor estimator

The maximum likelihood estimation of the standard quantization matrix requires the knowledge of the conditional probabilities $P(\hat{Q}^1_{ij}|Q^1_{ij}, Q^2_{ij})$ describing the accuracy of the primary quantization step detectors $\mathcal{F}_{Q^2_{ij}}$ for a given primary quantization step Q^1_{ij}. We calculated these probabilities on the images from the training set.

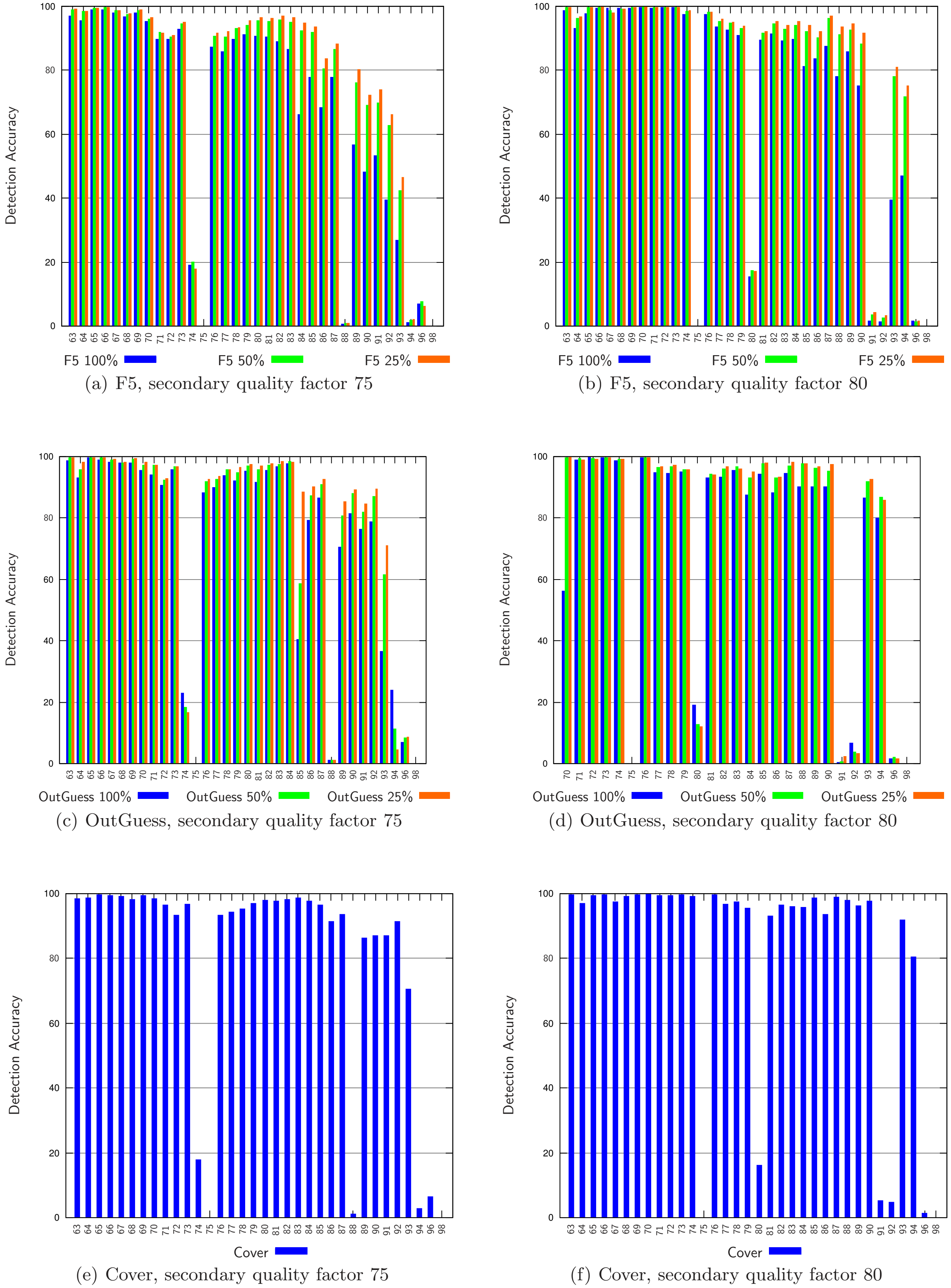
The accuracy of the estimator as function of the primary quality factor calculated on images from the testing set is shown in Figure 3. We can see that the estimator is robust with respect to steganographic modifications. The exception are images fully embedded by F5, which can be attributed to the shrinkage effect.

The accuracy exhibits some sharp drops all caused by the same mechanism (with the exception of images embedded by OutGuess with primary quality factor 75 and secondary quality factor 80). The primary quantization step estimators were constructed in such a way, that the divisor cases, when the primary quantization step Q^1_{ij} is a divisor of the secondary quantization step Q^2_{ij}, are detected by default as Q^2_{ij}. Let us now assume that Q and Q' are two primary quantization matrices for which

$$Q_{ij} \neq Q'_{ij} \Rightarrow Q_{ij}|Q^2_{ij} \text{ and } Q'_{ij}|Q^2_{ij}, \text{ for } (i,j) \in \mathcal{L}.$$

Let us further assume that for instance $\prod_{i,j\in\mathcal{L}} P(\hat{Q}^1_{ij}|Q_{ij}, Q^2_{ij}) > \prod_{i,j\in\mathcal{L}} P(\hat{Q}^1_{ij}|Q'_{ij}, Q^2_{ij})$. If we are analyzing an image with primary quantization matrix Q' and all quantization steps are detected correctly, the ML estimator will incorrectly output Q instead of Q' because Q has a larger likelihood. Even though this is a flaw of the estimator that cannot be easily overcome, it does not impact subsequent steganalysis, because the histograms of of the double-compressed DCT coefficients of the mode (i,j) are in both cases (quantized either by Q_{ij} or $Q^{'}_{ij}$) the same (the divisor effect).

We illustrate this phenomenon on an example of images with the primary quality factor 88 and the secondary quality factor 75. Most of the time, the primary quality factor is estimated as 89. We denote the quantization matrices corresponding to quality factors 89, 88, and 75 as $Q(89)$, $Q(88)$, and $Q(75)$, respectively. By examining the quantization steps of $Q(89)$ and $Q(88)$ for frequencies $(i,j) \in \mathcal{L}$, we observe that $Q(88)$ and $Q(89)$ only differ when $(i,j) = (0,1)$, in which case $Q^1_{01}(89) = 3$, $Q^1_{01}(88) = 2$, and $Q^2_{01}(75) =$



(a) F5, secondary quality factor 75

(b) F5, secondary quality factor 80

(c) OutGuess, secondary quality factor 75

(d) OutGuess, secondary quality factor 80

(e) Cover, secondary quality factor 75

(f) Cover, secondary quality factor 80

Figure 3. Accuracy of the primary quality factor estimator for secondary quality factors 75 and 80 on double-compressed cover images and images embedded with F5 and OutGuess algorithms. The x axis is the true primary quality factor.

6. If all primary quantization steps are correctly detected ($\hat{Q}_{01}^1$ is detected as 6), then the estimator of the primary quality factor will prefer the quality factor 89 over 88 because the conditional probability $P(\hat{Q}_{01}^1 = 6|Q_{01}^1 = 3, Q_{01}^2 = 3)$ is larger than $P(\hat{Q}_{01}^1 = 6|Q_{01}^1 = 2, Q_{01}^2 = 3)$ and all other involved probabilities are the same.

As for the case of the drop in the accuracy on images embedded by OutGuess with the primary quality factor 85 and the secondary quality factor 75, we strongly believe that it is due to the effect of embedding changes. We found out that most of the time, the primary quality factor was estimated as 84 instead of 85. The difference between the quantization matrices of these quality factors in $\mathcal{L}$ is for frequency $(0, 1)$, where $Q_{01}(84) = 4$ and $Q_{01}(85) = 3$. Because $Q_{01}(75) = 6$, this is not the case of the divisors discussed above. Figure 3(c) shows that the accuracy of the estimation improves as the length of the embedded message decreases, which supports our hypothesis about the influence of embedding.

6. CONCLUSION

The methods presented in this paper were developed primarily for blind steganalysis of JPEG images. They were designed to accurately detect double-compressed JPEG images and estimate their primary quantization tables not only for double-compressed cover images but also for double-compressed images with stego content.

We described one method for detection of double-compressed JPEG images that uses support vector machine classifiers with features derived from the first order statistics of individual DCT modes of low-frequency DCT coefficients. We also presented a Maximum Likelihood estimator of the primary quality factor in double-compressed JPEG images. This tool is essential for proper calibration of double-compressed images in both targeted[5]and blind steganalysis techniques.[13, 15]

The accuracy of the double-compression detector and the primary quality factor estimator was evaluated empirically on a testing database containing 1, 200, 000 single- and double-compressed images with combinations of 34 primary quality factors and 2 secondary quality factors (the default factors of F5 and OutGuess). Generally, the accuracy of both methods is better than 90% and the methods are robust against embedding operations. Even though there exist combinations of the primary and secondary quality factors, where the accuracy is low, their influence on subsequent steganalysis is negligible because those cases correspond to exactly the situations when the artifacts of double-compression are not present.

REFERENCES

1. JP Hide&Seek. http://linux01.gwdg.de/˜alatham/stego.html.
2. C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
3. Z. Fan and R. L. de Queiroz. Identification of bitmap compression history: JPEG detection and quantizer estimation. *IEEE Transactions on Image Processing*, 12(2):230–235, 2003.
4. J. Fridrich. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 67–81, 2005.
5. J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of *Lecture Notes in Computer Science*, pages 310–323, 2002.
6. J. Fridrich and J. Lukáš. Estimation of primary quantization matrix in double compressed JPEG images. In *Digital Forensic Research Workshop*, 2003.
7. D. Fu, Y. Q. Shi, and Q. Su. A generalized Benford's law for JPEG coefficients and its applications in image forensics. In E. Delp and P. W. Wong, editors, *Proceeedings of SPIE Electronic Imaging, Security and Watermarking of Multimedia Contents IX*, volume 6505, pages 1L1–1L11, 2007.
8. S. Hetzl and P. Mutzel. A graph-theoretic approach to steganography. In J. Dittmann et al., editor, *Communications and Multimedia Security. 9th IFIP TC-6 TC-11 International Conference*, volume 3677 of *Lecture Notes in Computer Science*, pages 119–128, 2005.
9. C. Hsu, C. Chang, and C. Lin. *A practical guide to support vector classification.* Department of Computer Science and Information Engineering, National Taiwan University, Taiwan. http://www.csie.ntu.edu.tw/˜cjlin/papers/guide/guide.pdf.

10. C. Hsu and C. Lin. A comparison of methods for multi-class support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001. http://citeseer.ist.psu.edu/hsu01comparison.html.
11. A. L. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.
12. W. Pennebaker and J. Mitchell. *JPEG: Still Image Data Compression Standard*. Van Norstrand Reinhold, 1993.
13. T. Pevný and J. Fridrich. Towards multi-class blind steganalyzer for JPEG images. In M. Barni, I. Cox, T. Kalker, and H. J. Kim, editors, *4th International Workshop on Digital Watermarking*, volume 3710 of *Lecture Notes in Computer Science*, pages 39–53. Springer-Verlag, Berlin, 2005.
14. T. Pevný and J. Fridrich. Determining the Stego Algorithm for JPEG Images. In *Special Issue of IEE Proceedings — Information Security*, volume 153, pages 75–139, 2006.
15. T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In E.J. Delp and P.W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, January 29–February 1*, volume 6505, pages 03–04, January 2007.
16. A.C. Popescu. *Statistical Tools for Digital Image Forensics*. PhD thesis, Dartmouth College, Hanover, New Hampshire, December 2004.
17. A.C. Popescu and H. Farid. Statistical tools for digital forensics. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 128–147, 2005.
18. N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, 2001.
19. P. Sallee. Model based steganography. In Kalker, I.J. Cox, and Yong Man Ro, editors, *International Workshop on Digital Watermarking*, volume 2939 of *Lecture Notes in Computer Science*, pages 154–167, 2004.
20. P. Sallee. Model-based methods for steganography and steganalysis. *Int. J. Image Graphics*, 5(1):167–190, 2005.
21. A. Westfeld. High capacity despite better steganalysis (F5 a steganographic algorithm). In I.S. Moskowitz, editor, *Information Hiding, 4th International Workshop*, volume 2137 of *Lecture Notes in Computer Science*, pages 289–302, 2001.