Detection of double-compression in JPEG images for applications in steganography

Tomáš Pevný and Jessica Fridrich

***Abstract*—This paper presents a method for detection of double JPEG compression and a maximum likelihood estimator of the primary quality factor. These methods are essential for construction of accurate targeted and blind steganalysis methods for JPEG images. The proposed methods use support vector machine classifiers with feature vectors formed by histograms of low-frequency DCT coefficients. The performance of the algorithms is compared to selected prior art.**

I. Introduction

Double-compression in JPEG images occurs when a JPEG image is decompressed to the spatial domain and than resaved with a different (secondary) quantization matrix. We call the first quantization matrix the primary quantization matrix. There are several reasons why we are interested in detecting double-compressed JPEG images and in the related problem of estimation of the primary quantization matrix.

First, detection of double compression is a forensic tool useful for recovery of the processing history. Double-compressed images are also frequently produced during image manipulation. By detecting the traces of recompression in individual image segments, we may be able to identify the forged region because the non-tampered part of the image will exhibit traces of double-compression [13], [17].

Second, some steganographic algorithms (Jsteg, F5 [18], OutGuess [14]) always decompress the cover JPEG image into the spatial domain before embedding. During embedding, the image is compressed again, usually with a default quantization matrix (F5 uses default quality factor 80, OutGuess 75). If the quantization matrix used during embedding differs from the original matrix, the resulting stego image is double-compressed. The statistics of DCT coefficients in double-compressed JPEG images may significantly differ from the statistics in single-compressed images. These differences negatively influence the accuracy of some steganalyzers developed under the assumption that the stego image has been compressed only once. This is especially true for steganalysis methods based on calibration [5], which is a process used to estimate macroscopic properties of the cover image from the stego image. If the steganographic method performs double compression, the calibration process has to be modified to mimic what happened during embedding, which requires the knowledge of the primary quantization matrix. Ignoring the effects of double-compression may lead to extremely inaccurate steganalysis results [5]. Thus, methods for detection of double-compression and estimation of the primary quantization matrix are essential for design of accurate steganalysis.

So far, all methods proposed for detection of double-compression and for estimation of the primary quantization matrix [7], [13], [6], [5] were designed under the assumption that the image under investigation is a cover image (not embedded). Because the act of embedding further modifies the statistics of DCT coefficients, there is a need for methods that can properly detect double compression in stego images and estimate the primary quantization matrix. Methods presented in this paper were developed to handle both cover and stego images, which makes them particularly suitable for applications in steganalysis.

This paper is organized as follows. We briefly review the basics of JPEG compression in Section II and continue with the discussion of prior art in Section III. Section IV describes the proposed methods for detecting double-compressed images and estimating the primary quantization matrix. In Section V, we show experimental results and compare them to prior art. We also discuss limitations of the proposed methods and their impact on subsequent steganalysis. Section VI contains conclusions.

II. Basics of JPEG Compression

The JPEG format is the most commonly used image format today. In this section, we only briefly outline the

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under the research grant number FA8750–04–1–0112. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U. S. Government.

Tomáš Pevný is a PhD. student at the Department of Computer Science, Binghamton University, NY 13902 USA (607-777-5689; fax: 607-777-5689; e-mail: pevnak@gmail.com)

Jessica Fridrich is a Professor at the Department of Electrical and Computer Engineering, Binghamton University, NY 13902 USA (607-777-6177; fax: 607-777-4464; e-mail: fridrich@binghamton.edu)

basic properties of the format that are relevant to our problem. A detailed description of the format can be found in [10].

During JPEG compression, the image is first divided into disjoint 8×8 pixel blocks $B_{rs},\ r,s = 0,\dots,7$. Each block is transformed using the Discrete Cosine Transformation (DCT)

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

where $w(0) = \frac{1}{\sqrt{2}}$ and $w(r > 0) = 1$. The DCT coefficients d_{ij} are then divided by quantization steps stored in the quantization matrix Q_{ij} and rounded to integers

$$D_{ij} = \mathrm{round}\left(\frac{d_{ij}}{Q_{ij}}\right),\ i,j \in \{0,\dots,7\}.$$

We denote the i,j-th DCT coefficient in the k-th block as $D^k_{ij},\ k \in \{1,\dots,l\}$, where l is the number of all 8×8 blocks in the image. The pair $(i,j) \in \{0,\dots,7\} \times \{0,\dots,7\}$ is called the *spatial frequency* (or *mode*) of the DCT coefficient. The JPEG compression finishes by ordering the quantized coefficients along a zig-zag path, encoding them, and finally applying lossless compression.

The decompression works in the opposite order. After reading the quantized DCT blocks from the JPEG file, each block of quantized DCT coefficients D is multiplied by the quantization matrix Q, $\hat{d}_{ij} = Q_{ij} \cdot D_{ij}$, and the Inverse Discrete Cosine Transformation (IDCT) is applied to $\hat{d}_{ij}$. The values are finally rounded to integers and truncated to a finite dynamic range (usually $[0, 255]$). The block of decompressed pixel values $\hat{B}$ is thus

$$\hat{B} = \mathrm{trunc}(\mathrm{round}(\mathrm{IDCT}(Q_{ij} \cdot D_{ij}))),\ i,j \in \{0,\dots,7\}.$$

Due to the rounding and truncation involved in compression and decompression, $\hat{B}$ will in general differ from the original block B.

Although there are not any standardized quantization matrices, most implementations of JPEG compression use a set of quantization matrices indexed by a *quality factor* from the set $\{1, 2, \dots, 100\}$. Theses matrices are used in the reference implementation[1] provided by the Independent JPEG Group. We refer to them as *standard* matrices.

We say that a JPEG image has been *double-compressed* if the JPEG compression was applied twice, each time with a different quantization matrix and with the same alignment with respect to the 8×8 grid. We call the first matrix Q^1 the *primary quantization matrix* and the second matrix Q^2 the *secondary quantization matrix*. Additionally, we say that a specific DCT *coefficient* D_{ij} was double-compressed if and only if $Q^1_{ij} \neq Q^2_{ij}$.

In a single-compressed JPEG image (only compressed with quantization matrix Q^1), the histogram of DCT coefficients for a fixed mode i,j, $D^k_{ij},\ k \in \{1,\dots.l\}$, is well-modeled with a quantized generalized Gaussian distribution [9]. When a single-compressed JPEG is decompressed to the spatial domain and compressed again with another quantization matrix Q^2, $Q^1 \neq Q^2$, the histograms of DCT coefficients no longer follow the generalized Gaussian distribution; they exhibit artifacts caused by double-compression.

Some of the most visible and robust artifacts in the histogram of DCT coefficients are *zeros* and *double-peaks* [6]. Zeros occur when some multiples of Q^2_{ij} in the double-compressed image are not present (see the odd multiples in Figure 1(b)), which occurs when $(Q^2_{ij} | Q^1_{ij}) \wedge (Q^1_{ij} \neq Q^2_{ij})$, where $a|b$ means "a divides b." Depending on the image and the values of the quantization steps, the zeros may take the form of local minimum rather than exact zeros.

A double peak occurs when a multiple of Q^1_{ij} falls in the middle of two multiples of Q^2_{ij} and no other multiple of Q^2_{ij} is closer. Formally, there exist integers $u, v \geq 0$, such that $uQ^1_{ij} = \frac{1}{2}\left((v-1)Q^2_{ij} + vQ^2_{ij}\right)$. In this case, the multiple uQ^1_{ij} contributes evenly to both $(v-1)Q^2_{ij}$ and vQ^2_{ij}. Figures 1(c,d) show examples of double-peaks occurring at multiples $v = 2, 5, 8, \dots$. For a more detailed description of the impact of double-compression on the DCT histogram, we refer to [6], [13].

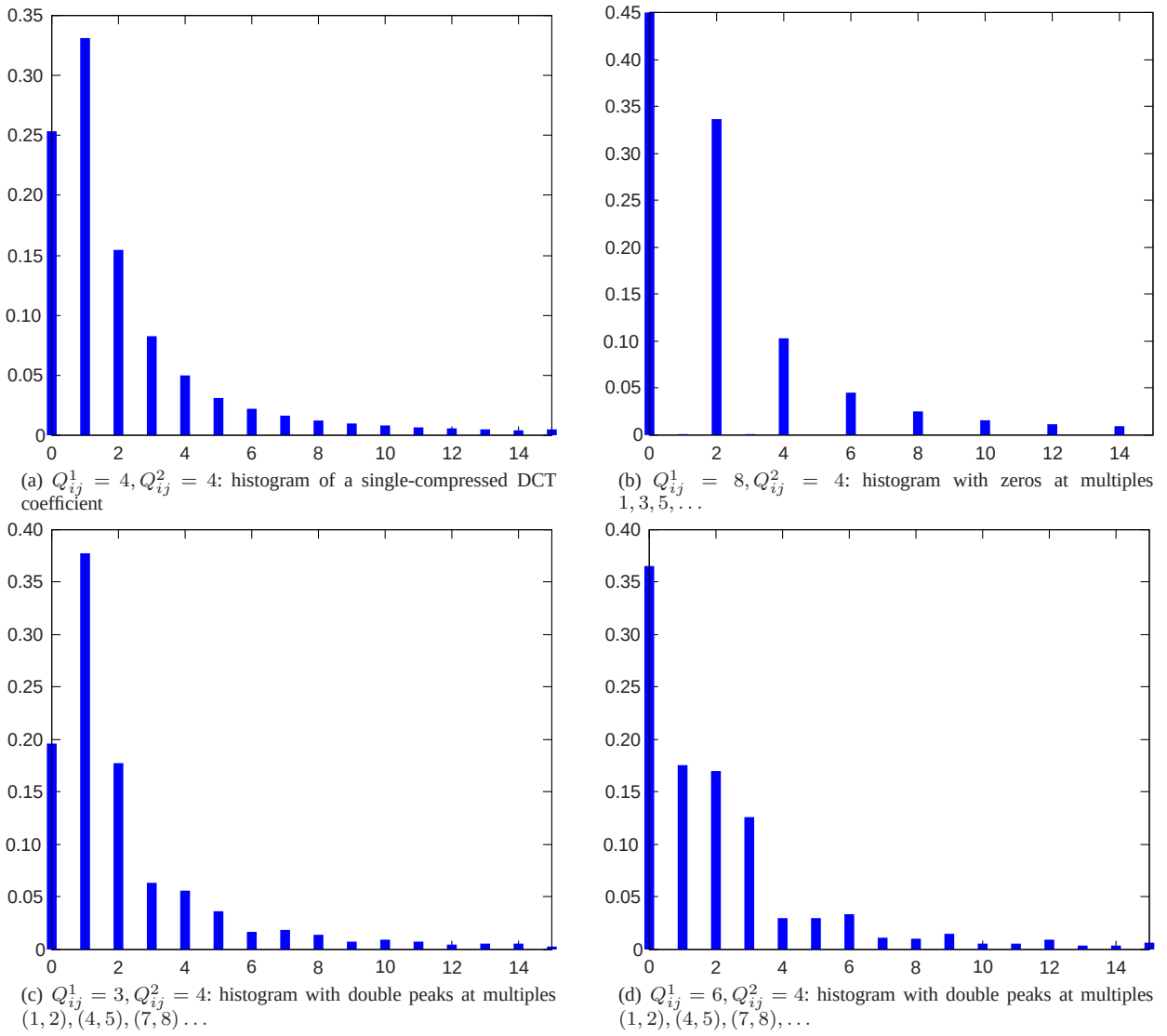
The examples shown in Figure 1 demonstrate that different combinations of primary and secondary quantization steps create distinct patterns in the histogram of DCT coefficients. Consequently, it is natural to use tools for pattern recognition to match histogram patterns to primary quantization steps. This idea was already exploited in [6] and is used in this paper as well.

III. Prior art

To the best of our knowledge, the first work dedicated to the problem of estimation of the primary quantization matrix in double-compressed images is due Fridrich et al. [6][2]. Instead of restoring the whole primary quantization matrix, the authors focused on estimation of the primary quantization steps Q^1_{ij} for low-frequency DCT coefficients $(i,j) \in \{(0,1),(1,1),(1,0)\}$, because the estimates for higher frequencies become progressively less reliable due to insufficient statistics. Three approaches were discussed. Two of them were based on matching the histograms of individual DCT coefficients of the inspected image with the histograms calculated from

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

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



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

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

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

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

Fig. 1. Examples of double-compression artifacts in histograms of absolute values of DCT coefficients for fixed mode $(0, 1)$.

estimates obtained by calibration [4], [5] followed by simulated double-compression. Both histogram matching approaches were outperformed by the third method that used a collection of neural networks. One neural network was trained for each value of the secondary quantization step of interest, $Q_{ij}^2 \in \{1, \ldots, 9\}$, to recognize the primary quantization step Q_{ij}^1 in the range $[2, 9]$, for $Q_{ij}^2 \in \{2, \ldots, 9\}$, and in the range $[1, 9]$, for $Q_{ij}^2 = 1$. All neural networks used the same input feature vector

$$x = \{h_{ij}(2), h_{ij}(3), \ldots, h_{ij}(15)\}, \tag{1}$$

where $h_{ij}(m)$ denotes the number of occurrences of $m \cdot Q_{ij}^2$ among absolute values of DCT coefficients D_{ij}^k in the inspected JPEG image.

$$h_{ij}(m) = \sum_{k=0}^{l} \delta\left(\left|D_{ij}^k\right| - m \cdot Q_{ij}^2\right), \tag{2}$$

where δ is the indicator function, $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ when $x \neq 0$. Note that the feature vector (1) does not include $h(0)$ and $h(1)$. The reported accuracy of this neural network detector 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 for frequencies $(i, j) \in \{(2, 0), (2, 1), (1, 2), (0, 2)\}$.

Recently, Shi et al. [7] proposed an idea for recovery of compression history of images based on the

observation that the distribution of the first (leading) digit[3] of DCT coefficients in single-compressed JPEG images of natural scenes follows the generalized Benford distribution

$$p(x) = N \cdot \log\left(1 + \frac{1}{s + x^q}\right) \; x \in \{1, \ldots, 9\}, \quad (3)$$

where q and s are free parameters and N is a normalization constant. This fact is used to estimate the quantization matrix Q of images previously JPEG compressed but currently stored in some other lossless image format (such as TIFF or PNG) in the following manner. The inspected image is compressed with different quality factors Q_t. When $Q_t \neq Q$, the image will be double-compressed and the generalized Benford law (3) will be violated. On the other hand, when $Q_t = Q$, the first digit of DCT coefficients will follow the generalized Benford distribution for some q and s, because the statistics of DCT coefficients is not affected by double-compression. The publication also proposes to use the histogram of the first digit of DCT coefficients as a feature set (further called the Benford feature set) for detection of double-compressed images. In Section V, we compare the Benford feature set to the feature set proposed in this paper by training and evaluating the classifiers on exactly the same database of images.

Popescu et al. [13], [12] presented another approach for detection of double-compression in JPEG images. The authors showed that repeated quantization introduces periodic artifacts into the histogram of DCT coefficients of each DCT mode. This periodicity can be detected as peaks in the Fourier domain. Unfortunately, we were unable to obtain the implementation from the authors and our own implementation produced results incompatible with those reported in [12], which prevented us from comparing the approaches.

IV. Proposed approach

The vast majority of steganographic techniques for JPEG images embed messages by directly manipulating the DCT coefficients. Most modern steganographic algorithms preserve first-order statistics and thus preserve the artifacts of double-compression. For example, Model Based Steganography preserves models of histograms of each DCT mode, OutGuess preserves the global histogram of all DCT coefficients (but not necessarily the histograms of individual DCT modes). F5 does change the histogram but in a smooth manner, making it more spiky. This, too, would preserve the qualitative character of double-compression artifacts. Thus, it is reasonable to expect that double-compression can be detected from stego images using pattern recognition methods.

Many steganalytic methods need to know the compression history of a given stego image to produce accurate results. Steganalysis based on calibration (estimation of the cover image) is especially vulnerable as it may produce completely misleading results when the effect of double-compression is not accounted for. Reliable detection of double-compression is also important for so called multi-classifiers that attempt to not only detect the presence of a secret message but also classify the stego image to a known steganographic method. For example, the blind multi-classifier described in [11] consists of a double-compression detector and two separate classifiers—one trained for single compressed images and one specially built for double-compressed images. The double-compression detector thus serves as a pre-classification. When it decides that an image has been double-compressed, it already points to those methods that can produce such images (F5 and OutGuess). Mistakenly detecting a single-compressed image as double-compressed may thus introduce large classification errors for the entire multi-classifier, because it can now only answer either cover, F5, or OutGuess. What is needed is a double-compression detector with a low probability of false positives, which means low probability of detecting a single-compressed image as double-compressed. False negatives (detecting a double-compressed image as single-compressed) are much less serious because when double-compressed images are misclassified due to lack of presence of double-compression artifacts, the effect of double-compression on steganalysis is also small.

The problem of double-compression detection could be considered as a sub-problem of the primary quantization matrix estimation. We could detect whether the image was double-compressed by comparing the estimated primary quantization matrix with the secondary quantization matrix. Unfortunately, this naïve approach is not very accurate. Better performance can be achieved with a separate double-compression detector (further called the DC detector) followed by the primary quality factor estimator (the PQF estimator) applied only to images classified as double-compressed.

The positive experience with a combination of classification tools and features formed by histograms of multiples of quantization steps in [6] steered our attention in this direction. Because of the problem with insufficient statistics for high-frequency DCT coefficients mentioned in the previous section, we also limit the set of DCT frequencies used by both the DC detector and the PQF estimator to the set

[3]The first digit is understood as the first *valid* digit in the decimal representation of the number. The numbers are *not* padded by zeros to have the same number of digits. For example 2 and 25 have the same first digit 2.

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

Before we describe the details of our method, we briefly discuss another possibility to estimate the primary quantization matrix from the statistics of DCT coefficients D_{ij} even though we do not pursue this method in this paper. We could model the distribution of DCT coefficients for a fixed spatial frequency in the single-compressed image using a parametric model (e.g., generalized Gaussian) and estimate the primary quantization step, together with the nuisance model parameters, using the Maximum Likelihood (ML) principle and avoid using classification altogether. While this choice does sound tempting, the distribution of DCT coefficients may be affected by embedding and thus the ML estimator may produce inaccurate results because of a model mismatch. Indeed, the F5 algorithm makes the distribution of DCT coefficients more spiky by increasing the number of zeros due to shrinkage. Even OutGuess modifies the distribution of coefficients for *individual* DCT modes (it only preserves the *global* histogram).

A. Detection of double-compression

The double-compression detector is implemented using a soft-margin support vector machine ($C-$SVM) with the Gaussian kernel [2] $k(x,y) = \exp(-\gamma\|x-y\|^2)$. Its feature vector x consists of histograms (2) for spatial frequencies from the set $\mathcal{L}$. Formally,

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

where C_{ij} are normalization constants $\left(C_{ij} = \sum_{m=0}^{15} h_{ij}(m)\right)$. The dimension of this feature set (further called the Multiple-counting feature set) is $16 \times 9 = 144$.

Because the DC detector is a binary classifier, it is easy to adjust its bias towards one class. As already mentioned in the introduction to this section, this feature is important for applications in steganalysis.

B. Detector of primary quantization steps

The double-compression detector described in the previous section generates binary output—the image is either single or double-compressed. In this section, we introduce a method for detecting the individual primary quantization steps and then in Section IV-C, we explain the process of matching the detected quantization steps to the closest standard matrix.

We only detect the primary quantization steps for spatial frequencies from the set $\mathcal{L}$. This detector consists of a collection of SVM-based multi-classifiers $\mathcal{F}_{Q_{ij}^2}$ for each value of the secondary quantization step Q_{ij}^2. In our experiments, we created five multi-classifiers for the secondary quantization steps $Q_{ij}^2 \in \{4,5,6,7,8\}$, because this is the range of quantization steps for spatial frequencies $\mathcal{L}$ from secondary quantization matrices with quality factors 75 and 80 (the default quality factors in F5 and OutGuess). Table I shows the primary quantization steps detectable by the multi-classifiers for each secondary quantization step and the number of SVMs in the multi-classifier.

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 I
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.

We note that it is possible to detect other primary quantization steps Q_{ij}^1. To this end, one would have to prepare examples of images with required combinations of the primary Q_{ij}^1 and secondary quantization Q_{ij}^2 steps and appropriately extend the multi-classifier $\mathcal{F}_{Q_{ij}^2}$ (training a set of binary classifiers as described below).

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

$$x = \frac{1}{C}(h_{ij}(0), h_{ij}(1), \ldots, h_{ij}(15)), \tag{4}$$

where C is a normalization constant chosen so that $\sum_{m=0}^{15} x_m = 1$. The multi-classifier $\mathcal{F}_{Q_{ij}^2}$ consists of a collection of binary classifiers. Since there is one binary classifier for every combination of two different primary quantization steps, the number of binary classifiers is $\binom{n}{2}$, where n is the number of classes. For example, for the secondary quantization step 4, we classify into $n = 6$ classes, for which we need $\binom{6}{2} = 15$ binary classifiers. During classification, the feature vector (4) is presented to all binary classifiers. Every binary classifier gives vote to one primary quantization step. At the end, the votes are counted and the quantization step with most votes is selected as the winner. Ties are resolved by selecting the primary quantization step randomly from the set of steps with the highest number of votes to avoid creating a bias towards one class. All binary classifiers are soft-margin Support Vector Machines ($C-$SVM) with the Gaussian kernel.

Note that the feature vector (4) cannot distinguish

between the following three cases: Q_{ij}^1 is a divisor of Q_{ij}^2, $Q_{ij}^1 = 1$, and $Q_{ij}^1 = Q_{ij}^2$. Thus, we classify all these cases into one common class $Q_{ij}^1 = Q_{ij}^2$. This phenomenon imposes a fundamental limitation on the performance of the detector. Fortunately, the double-compressed image in all these three cases does not exhibit any discernible traces of double-compression, and hence influences steganalysis in a negligible manner. In other words, our failure to distinguish between these cases is not detrimental for steganalysis.

C. Matching the closest standard quantization matrix

The primary quantization step detector presented in the previous section only estimates the primary quantization steps for a small set of spatial frequencies from the set $\mathcal{L}$. Since we wish to recover the whole quantization matrix (e.g., in order to carry out calibration in steganalysis), a procedure is needed that will find the whole primary quantization matrix. Moreover, because the detection will sometimes produce incorrect values of the primary quantization steps, the procedure should reveal such outliers and replace them with correct values. We achieve both tasks by finding the closest standard quantization matrix using a Maximum Likelihood estimator.

Denoting the detected and the true primary quantization steps as $\hat{Q}_{ij}^1$ and Q_{ij}^1, respectively, the closest standard quantization matrix can be obtained using the ML estimator

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

where $\mathcal{T}$ is the set of standard quantization matrices. The set $\mathcal{T}$ can be modified to incorporate available side information (for example some camera manufacturers use customized quantization matrices). The value $P(\hat{Q}_{ij}^1 | Q_{ij}^1, Q_{ij}^2)$ is the probability that the classifier detects the primary quantization step $\hat{Q}_{ij}^1$ when the correct primary quantization step is Q_{ij}^1 and the secondary quantization step is Q_{ij}^2. These probabilities can be empirically estimated from images used for training the detector.

We note that it is possible to incorporate a priori knowledge about the distribution of primary quantization tables into the estimation procedure and switch to a MAP estimator. This a priori information could be obtained by crawling the web and collecting the statistics about the JPEG quality tables. In this paper, however, we do not pursue this approach.

V. Experimental results

In this section, we present experimental results and compare them to selected prior art. All results in this section, including the prior art evaluation, were calculated on a database created from 6006 raw images. Before conducting any experiments, the images were divided into a training subset containing 3500 raw images and a testing subset containing 2506 raw images. This allowed us to estimate the performance on images that were never used in any form in the training phase. The testing subset contains images taken by different cameras and photographers.

The double-compressed stego images were created by OutGuess and F5. We embedded message lengths 100%, 50%, and 25% of embedding capacity for each algorithm and image. These two steganographic algorithms were selected because their implementations produce double-compressed images. The double-compressed images were prepared with 34 different primary quality factors $\mathcal{Q}_{34} = \{63, 64, \ldots, 93, 94, 96, 98\}$ and with two different secondary quality factors: 75, which is the default quality factor of OutGuess, and 80, the default quality factor of F5.

Because we need to test the performance of the DC detector also on single-compressed images to evaluate its false positive rate, we also prepared single-compressed images with quality factors 75 and 80 embedded by the following steganographic algorithms: F5 [18], Model Based Steganography without [15] (MBS1) and with [16] deblocking (MBS2), JP Hide&Seek [1], OutGuess [14], and Steghide [8]. The embedded message length was chosen as 100%, 50%, and 25% of the embedding capacity for each algorithm. All MBS2 images were embedded only with 30% of the capacity of MBS1, because during embedding of longer messages the deblocking part of MBS2 usually fails.

The resulting database, which contains both double- and single-compressed images, contains cover images with the same combinations of primary and secondary quality factors as the stego images. The total number of processed images was $34 \times 2 \times 7 \times (3500 + 2506) + 17 \times (3500 + 2506) \approx 3,000,000$.

A. Double-compression detector

In this section, we describe the details for constructing the detector of double-compressed images with secondary quality factors 75 and 80 (see Section IV-A). Due to extensive computational complexity, instead of training a general double-compression detector for both quality factors, we decided to train a special double-compression detector for each secondary quality factor (The complexity of training a $C-$SVM is $O(N^3)$, where N is the number of examples.).

All classifiers were implemented using the soft-margin $C-$SVM and were trained on 10000 examples of single-compressed images (cover images and images embedded by the six aforementioned steganographic algorithms) and on 10000 examples of double-compressed images

(cover images and images embedded by F5 and OutGuess). The hyper-parameters C and γ were determined by a grid-search on the multiplicative grid

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

combined with $5-$fold cross-validation.

Figure 2 shows the accuracy of the DC detector on double-compressed JPEG images from the testing set. We can see that the accuracy on cover images and images embedded by OutGuess is very good. The accuracy on F5 images is worse, especially on images containing longer messages. We attribute this loss of accuracy to the fact that F5 alters the shape of histograms of DCT coefficients. As the primary quality factor increases, artifacts of double-compression are becoming more subtle and the accuracy of the detector decreases, which is to be expected.

In Figure 2, we can observe sharp drops in the accuracy of the detector on images with primary quality factors 96 and 98, and on images with primary quality factor 74 and secondary quality factor 75. These sharp drops correspond to situations when the histograms of DCT coefficients are not affected by double-compression—all primary quantization steps for frequencies from $\mathcal{L}$ are divisors of the secondary quantization steps. The quantization steps for all 9 frequencies from $\mathcal{L}$ for standard matrices with quality factors 96 and 98 are all ones. Similarly, the quantization steps in the standard quantization matrices with quality factors 74 and 75 satisfy $Q_{ij}(74) = Q_{ij}(75), (i, j) \in \mathcal{L}$. Consequently, the decision of the detector is correct, since in these cases, the DCT coefficients in $\mathcal{L}$ are not double-compressed. We note that we avoided using images with these combinations of quality factors in the training set.

Figure 3 shows the accuracy of the double-compression detector on single-compressed JPEG images embedded by various steganographic algorithms. Almost all of the tested steganographic algorithms preserve the histogram of DCT coefficients, which helps the detector to maintain its good accuracy. The only exception is F5 already commented upon above.

B. Benford features

In Section III, we mentioned an approach proposed by Shi et al. [7] to use the histogram of the distribution of the first digit of DCT coefficients as a feature vector for a classifier detecting double-compression. In order to compare Benford features to Multiple-counting features described in Section IV-A, we prepared two $C-$SVM classifiers—one for each feature set. Both classifiers were trained on cover images with the (secondary) quality factor 75. The size of the training set was 6800 examples.

	Benford	Multiple
Single-compressed	61.74%	98.64%
Double-compressed	30.91%	97.11%

TABLE II
ACCURACY OF DOUBLE-COMPRESSION DETECTOR EMPLOYING BENFORD AND MULTIPLE-COUNTING FEATURES. DETECTORS ARE TRAINED AND TESTED ON COVER IMAGES ONLY.

Table II shows the detection accuracy of both classifiers calculated on images from the testing set. We excluded double-compressed images with primary quality factors 74, 96, and 98 because DCT coefficients with spatial frequencies in $\mathcal{L}$ are not technically double-compressed in those cases. Table II shows that while the performance of the Benford features on our database of cover images is close to random guessing with bias towards the single-compressed class, the accuracy of Multiple-counting features is about 98%.

C. Estimation of primary quantization coefficients

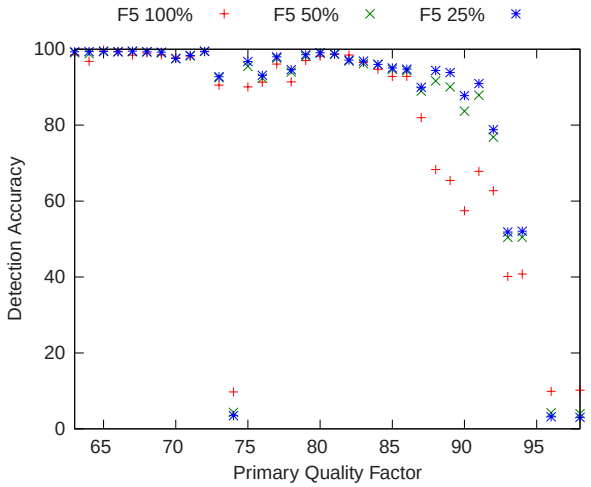
This section presents experimental results of the detector of the primary quantization steps. As described in Section IV-B, the detector is implemented by a collection of "max-wins" multi-classifiers, where each multi-classifier consists of the set of soft-margin Support Vector Machines ($C-$SVM) with the Gaussian kernel. The training set for each $C-$SVM contained 20000 examples—10000 from each class. The hyper-parameters C and γ were estimated by means of a $5-$fold cross-validation on the multiplicative grid

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

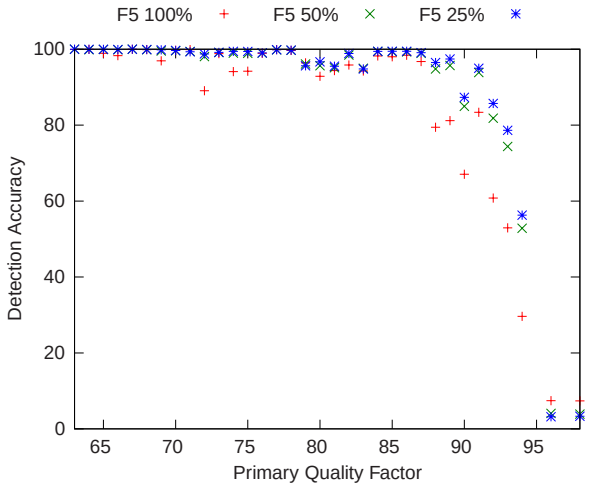
For training, we used C and γ corresponding to the point with the least cross-validation error.

Tables III and IV compare the accuracy of the SVM-based primary quantization step detector with the Neural Network (NN) detector[4] from [6] on images from the testing set. The comparison is done for the secondary quantization steps 4, 5, 6, 7, and 8. The NN detector detects only the quantization steps in the range $[1, 9]$. We have to point out that while the SVM detector was trained on cover and stego images, the NN detector was trained on cover images only. Because of this difference, we present the results on a mixed database of cover and stego images (Table IV) and on cover images only (Table III). In most cases, the SVM based detector outperformed the NN detector. The rare occasions when the NN detector gave better results correspond to the situation when the primary quantization step was a divisor of the secondary step. As explained in Section IV-B, incorrect primary step detection in these cases has virtually no influence on steganalysis.

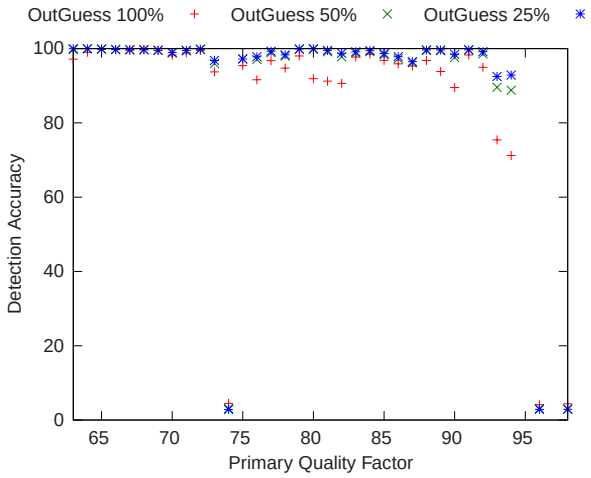
[4]The trained detector was kindly provided to us by the authors of [6].



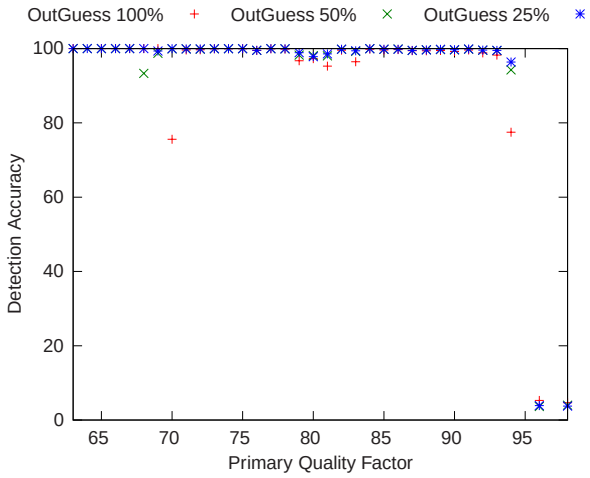
(a) F5, secondary quality factor 75



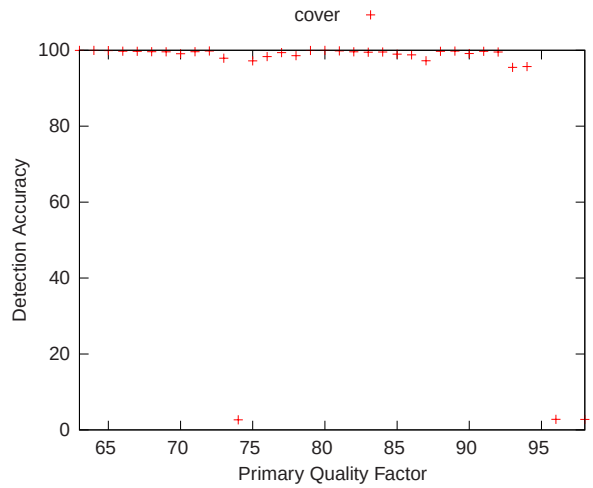
(b) F5, secondary quality factor 80



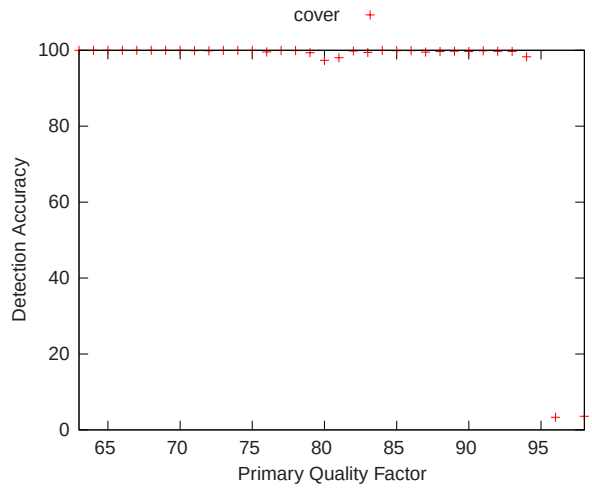
(c) OutGuess, secondary quality factor 75



(d) OutGuess, secondary quality factor 80



(e) Cover, secondary quality factor 75



(f) Cover, secondary quality factor 80

Fig. 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. Images with primary quality factor equal to the secondary quality factor are not double-compressed, which means that in this case, the correct answer of the detector is single-compressed. Graphs are drawn with respect to the primary quality factor.

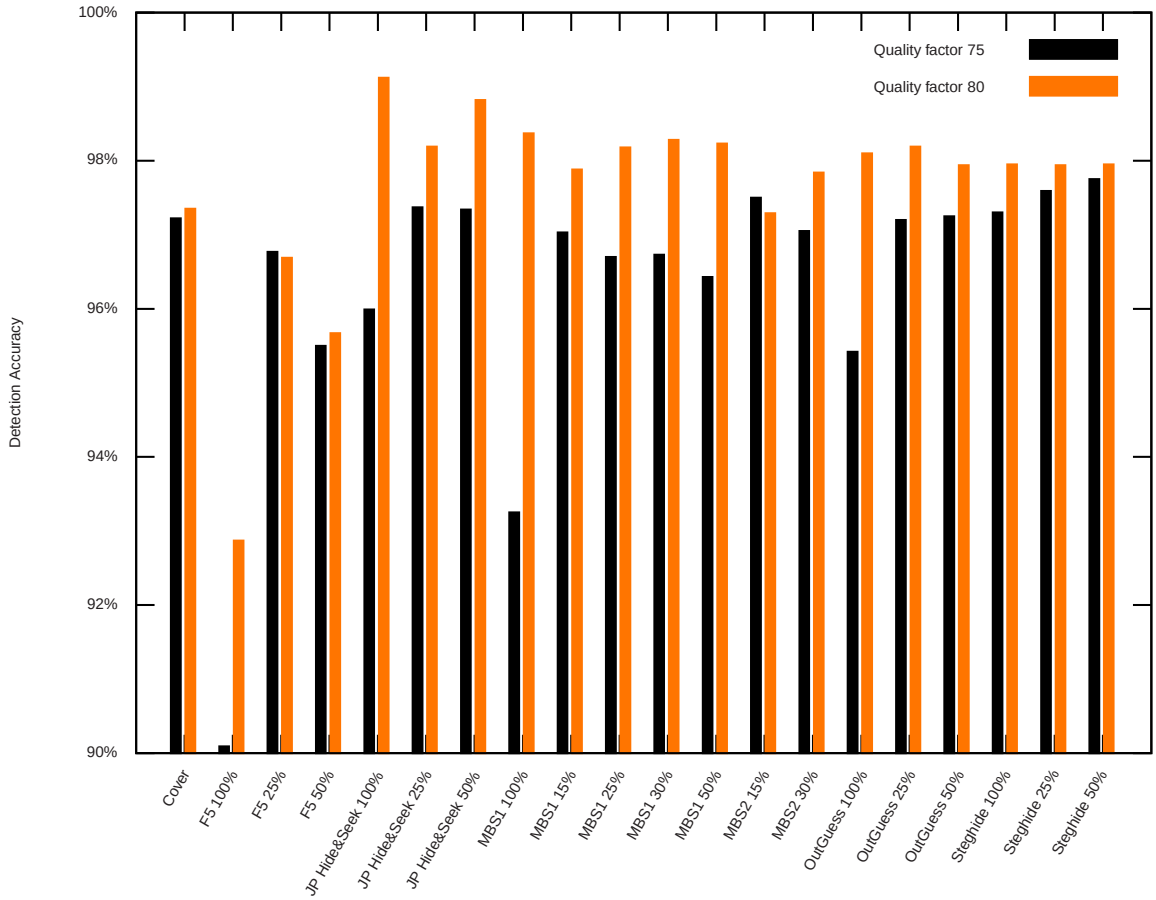
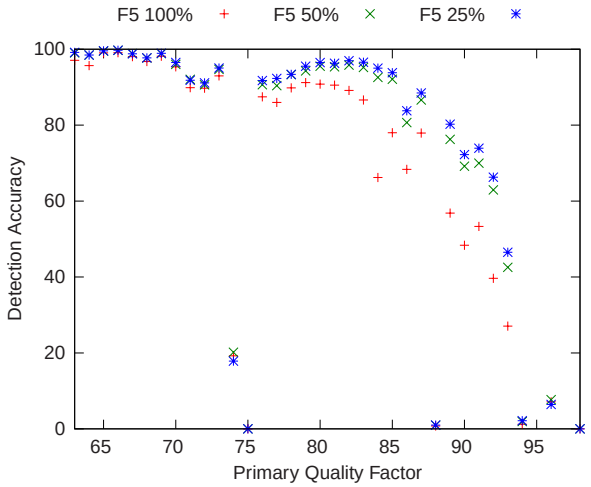


Fig. 3. Accuracy of the double-compression detector on single-compressed JPEG images with quality factors 75 and 80. Note that the range of Y axis is 90% to 100%.

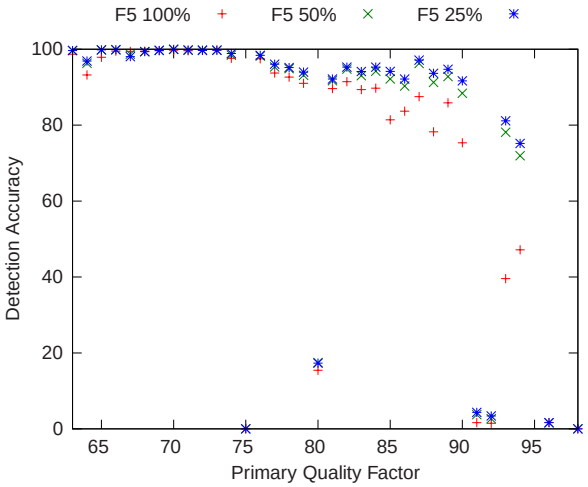
SQS	4		5		6		7		8	
PQS	SVM	NN	SVM	NN	SVM	NN	SVM	NN	SVM	NN
1	96.03%	**98.69%**	85.22%	**87.46%**	**92.47%**	91.41%	79.85%	**95.76%**	67.81%	**90.97%**
2	96.23%	**98.63%**	**95.32%**	74.79%	**92.12%**	91.82%	**86.38%**	74.18%	69.37%	**91.47%**
3	**98.85%**	96.95%	**98.75%**	87.29%	**93.64%**	90.15%	**88.24%**	77.88%	**77.80%**	52.81%
4	95.70%	**98.75%**	**96.83%**	94.43%	**98.66%**	90.32%	**90.75%**	77.05%	71.14%	**91.86%**
5	**99.80%**	95.08%	84.30%	**86.45%**	**95.32%**	91.06%	**96.29%**	81.47%	**95.13%**	65.28%
6	**99.15%**	98.44%	**99.47%**	85.72%	**91.62%**	91.07%	**89.95%**	89.90%	90.75%	**94.09%**
7	**99.51%**	98.91%	**99.45%**	90.38%	**98.54%**	96.47%	80.04%	**95.66%**	**83.67%**	59.00%
8	**99.84%**	99.80%	**98.89%**	97.01%	**99.54%**	96.07%	**95.40%**	88.23%	67.02%	**91.22%**
9	—	—	98.35%	**98.69%**	**97.23%**	95.84%	**98.65%**	84.61%	**92.24%**	81.35%
10	—	—	**99.72%**	—	**99.85%**	—	**98.73%**	—	**93.64%**	—
11	—	—	—	—	**92.01%**	—	—	—	**97.91%**	—
12	—	—	—	—	**97.38%**	—	—	—	**99.08%**	—

TABLE III

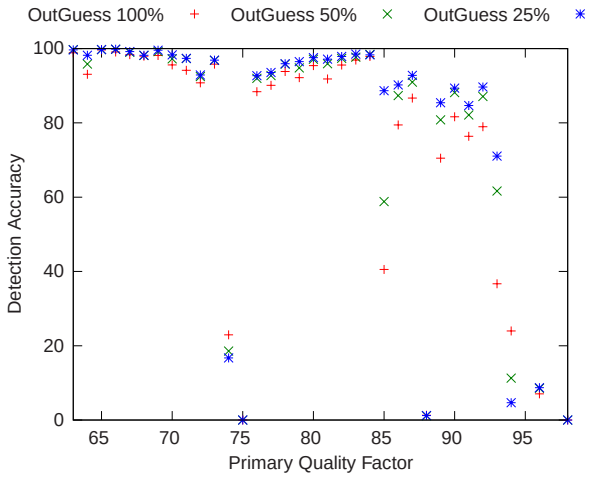
ACCURACY OF NEURAL NETWORK (NN) AND SUPPORT VECTOR MACHINE (SVM) PRIMARY QUANTIZATION STEPS DETECTORS ON *COVER* IMAGES FROM THE TESTING SET. PQS AND SQS STAND FOR PRIMARY AND SECONDARY QUANTIZATION STEPS, RESPECTIVELY.



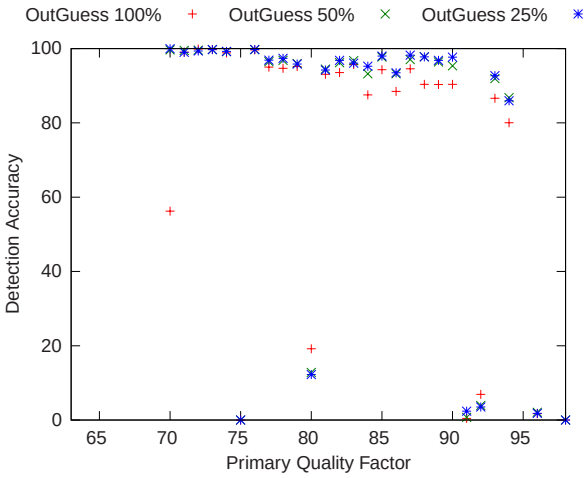
(a) F5, secondary quality factor 75



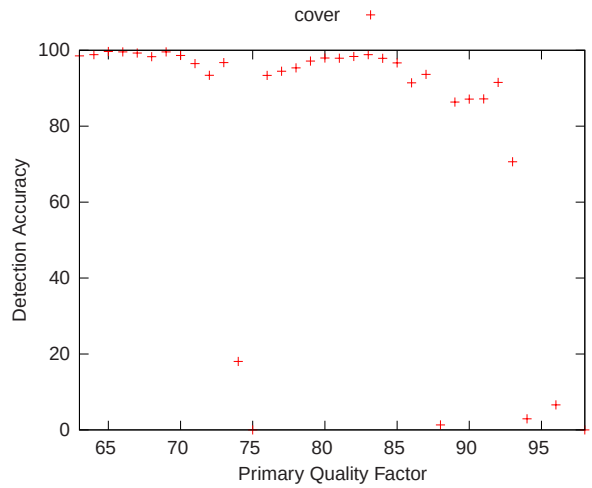
(b) F5, secondary quality factor 80



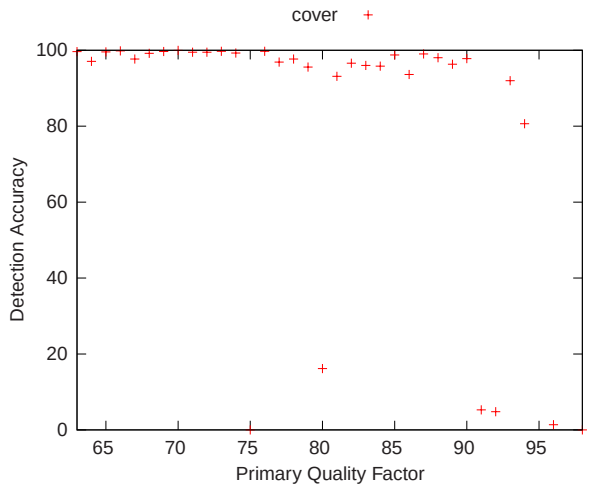
(c) OutGuess, secondary quality factor 75



(d) OutGuess, secondary quality factor 80



(e) Cover, secondary quality factor 75



(f) Cover, secondary quality factor 80

Fig. 4. Accuracy of primary quality factor estimator for secondary quality factors 75 and 80 on double-compressed cover images and images embedded with F5 and OutGuess algorithms. Graphs are drawn with respect to the true primary quality factor.

SQS	4		5		6		7		8	
PQS	SVM	NN	SVM	NN	SVM	NN	SVM	NN	SVM	NN
1	95.24%	**98.56%**	86.75%	**87.92%**	91.01%	**90.95%**	78.74%	**95.79%**	66.03%	**90.31%**
2	95.51%	**98.59%**	**84.17%**	45.16%	90.99%	**91.67%**	**65.32%**	44.59%	66.64%	**90.31%**
3	**95.19%**	67.41%	**94.15%**	59.20%	**92.43%**	89.78%	**81.99%**	52.19%	**72.06%**	36.79%
4	94.23%	**98.62%**	**95.12%**	71.84%	**94.62%**	59.41%	**83.46%**	52.16%	70.69%	**90.67%**
5	**99.43%**	78.47%	83.99%	**86.32%**	**94.03%**	70.68%	**91.33%**	51.50%	**87.67%**	40.48%
6	**99.36%**	76.86%	**98.26%**	66.45%	88.40%	**89.93%**	**85.02%**	70.87%	**83.60%**	68.27%
7	**99.58%**	61.72%	**99.47%**	61.72%	**97.20%**	81.09%	77.21%	**94.35%**	**79.22%**	42.00%
8	**99.40%**	72.16%	**98.92%**	70.29%	**99.40%**	59.26%	**93.78%**	58.32%	63.42%	**88.89%**
9	—	—	**97.56%**	72.37%	**97.79%**	80.80%	**97.39%**	58.90%	**87.50%**	58.26%
10	—	—	**99.23%**	—	**99.58%**	—	**98.75%**	—	**91.17%**	—
11	—	—	—	—	**90.45%**	—	—	—	**96.98%**	—
12	—	—	—	—	**96.08%**	—	—	—	**98.87%**	—

TABLE IV
ACCURACY OF NEURAL NETWORK (NN) AND SUPPORT VECTOR MACHINE (SVM) PRIMARY QUANTIZATION STEPS DETECTORS ON *COVER* AND *STEGO* IMAGES FROM THE TESTING SET. PQS AND SQS STAND FOR PRIMARY AND SECONDARY QUANTIZATION STEPS, RESPECTIVELY.

D. Estimation of the standard quantization matrix

The estimator 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 detector of the primary quantization steps. These probabilities were evaluated empirically from images from the training set, as noted in Section IV-C.

Figure 4 shows the accuracy calculated on images from the testing set as a function of the true primary quality factor. We conclude that the accuracy is not much affected by embedding. The detection on stego images embedded by F5 is worse (especially on fully embedded images) due to F5's influence on the histogram.

All sharp drops in accuracy have the same cause, with the exception of images embedded by OutGuess with primary quality factor 75 and secondary quality factor 80. We will discuss this case later. As explained in Section IV-B, the cases when the primary quantization step Q^1_{ij} is a divisor of the secondary quantization step Q^2_{ij}, the primary quantization step is detected by default as Q^2_{ij}. Let us 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 $\prod_{i,j\in\mathcal{L}} P(\hat{Q}^1_{ij}|Q_{ij}, Q^2_{ij}) > \prod_{i,j\in\mathcal{L}} P(\hat{Q}^1_{ij}|Q'_{ij}, Q^2_{ij})$. When detecting images with primary quantization matrix Q' (if all quantization steps are detected correctly), the ML estimator will incorrectly output Q instead of Q' because Q has a larger likelihood. This failure is, fortunately, not going to impact subsequent steganalysis because when the primary quantization steps are divisors of the secondary quantization step, the impact of double-compression is negligible.

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 $Q^2_{01}(75) = 6$. If all primary quantization steps are correctly detected ($\hat{Q}^1_{01}$ is detected as 6), then the estimator of the primary quality factor will prefer the quality factor 89 over 88 because the conditional probability $P(\hat{Q}^1_{01} = 6|Q^1_{01} = 3, Q^2_{01} = 3)$ is larger than $P(\hat{Q}^1_{01} = 6|Q^1_{01} = 2, Q^2_{01} = 3)$ (see Table IV) and all other involved probabilities are the same.

The drop in accuracy on images embedded by OutGuess with the primary quality factor 85 and the secondary quality factor 75 is caused by the effect of embedding. The majority of incorrectly estimated images have the primary quality factor estimated as 84 instead of 85. The difference between the quantization matrices $Q(84)$ and $Q(85)$ 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 divisors discussed above. From Figure 4(c), we see that the accuracy of estimation improves on images with shorter messages, which confirms our hypothesis about the influence of embedding.

Table V shows the average decrease in the detection accuracy when the PQF estimator was first applied only to stego images and then only to cover images. Because F5 changes the histogram of DCT coefficients, the accuracy of the PQF estimator is worse for F5 embedded images than for OutGuess, which preserves the global histogram. The accuracy of the PQF estimator is expected to be even lower for steganographic techniques that significantly modify the histograms. It is unlikely, however, that such techniques will ever be developed

	SQF 75		SQF 80	
Algorithm	Mean	Std	Mean	Std
F5 100%	11.07	14.24	6.70	11.17
F5 50%	4.82	7.52	2.14	3.33
F5 25%	3.82	6.30	1.39	2.40
OutGuess 100%	5.12	12.08	3.76	8.55
OutGuess 50%	2.06	6.91	0.36	1.84
OutGuess 25%	0.37	1.67	0.11	1.48

TABLE V
MEAN AND STANDARD DEVIATION OF THE DROP IN ACCURACY OF THE PRIMARY QUALITY FACTOR ESTIMATOR WHEN APPLYING IT ONLY TO STEGO IMAGES AND ONLY TO COVER IMAGES.

because steganography significantly disturbing the first order statistics would likely be detectable using other means.

VI. CONCLUSION

The contribution of this paper is two-fold. First, we presented a reliable method for detection of double-compressed JPEG images. It is based on classification using support vector machines with features derived from the first order statistics of individual DCT modes of low-frequency DCT coefficients. An important feature of the proposed method is its ability to detect double-compression not only for cover images but also for images processed using steganographic algorithms. By comparing our method to prior art, we showed that the proposed solution offers higher accuracy.

Second, we built a maximum likelihood estimator of the primary quality factor in double-compressed JPEG images. Since the main application is steganalysis, the estimator was constructed to work for both cover and stego images. We evaluated the accuracy of the estimator on a large test of JPEG images with 34 primary quality factors and 2 secondary quality factors (the default factors of F5 and OutGuess). Generally, the accuracy is better than 90% and is not much affected by embedding operations. There exist combinations of the primary and secondary quality factors, where the accuracy is low. They all correspond to situations when the effects of double-compression are negligible and thus the failures do not influence subsequent steganalysis. To the best of our knowledge, this is the first complete solution to the problem of estimation of the primary quality factor in double-compressed JPEG images in the context of steganalysis.

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] A. L. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[10] W. Pennebaker and J. Mitchell. *JPEG: Still Image Data Compression Standard*. Van Norstrand Reinhold, 1993.

[11] 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.

[12] A.C. Popescu. *Statistical Tools for Digital Image Forensics*. PhD thesis, Dartmouth College, Hanover, New Hampshire, December 2004.

[13] 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.

[14] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, 2001.

[15] 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.

[16] P. Sallee. Model-based methods for steganography and steganalysis. *Int. J. Image Graphics*, 5(1):167–190, 2005.

[17] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double MPEG compression. In J. Dittmann and J. Fridrich, editors, *Proceedings ACM Multimedia and Security Workshop*, pages 37–47. ACM Press, New York.

[18] 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.