STATISTICAL MODELS FOR IMAGE STEGANOGRAPHY

EXPLAINING AND REPLACING HEURISTICS

BY

JAN BUTORA

MS, Charles University, Prague, 2017

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate School of Binghamton University
State University of New York
2021

Accepted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering
in the Graduate School of Binghamton University
State University of New York
May 7st, 2021

Jessica Fridrich, Chair and Faculty Advisor
Department of Electrical and Computer Engineering, Binghamton University

Mark Fowler, Member
Department of Electrical and Computer Engineering, Binghamton University

Scott Craver, Member
Department of Electrical and Computer Engineering, Binghamton University

Ronald Miles, Outside Examiner
Department of Mechanical Engineering, Binghamton University

Abstract

The steganographic field is nowadays dominated by heuristic approaches for data hiding. While there exist a few model-based steganographic algorithms designed to minimize statistical detectability of the underlying model, many more algorithms based on costs of changing a specific pixel or a DCT coefficient have been over the last decade introduced. These costs are purely heuristic, as they are designed with feedback from detectors implemented as machine learning classifiers. For this reason, there is no apparent relation to statistical detectability, even though in practice they provide comparable security to model-based algorithms. Clearly, the security of such algorithms stands only on the assumption, that the detector used to assess the security, is the best one possible. Such assumption is of course completely unrealistic.

Similarly, steganalysis is mainly implemented with empirical machine learning detectors, which use hand-crafted features computed from images or as deep learning detectors - convolutional neural networks. The biggest drawback of this approach is, that the steganalyst, even though having a very good detection power, has very little to no knowledge about what part of the image or the embedding algorithm contributes to the detection, because the detector is used as a black box.

In this work, we will try to leave the heuristics behind and go towards statistical models. First, we introduce statistical models for current heuristic algorithms, which helps us understand and predict their security trends. Furthemore this allows us to improve the security of such algorithms. Next, we focus on steganalysis exploiting universal properties of JPEG images. Under certain realistic conditions, this leads to a very powerful attack against any steganography, because embedding even a very small secret message breaks the statistical model. Lastly, we show how we can improve security of JPEG compressed images through additional compression.

Acknowledgements

I would like to thank my Ph.D. advisor, Jessica Fridrich, for her support and guidance through my studies. She created a research environment that motivated me to push and improve myself professionally but also a friendly atmosphere which made me feel welcome and appreciated. Our long discussion, that would occasionally go a little sideways, made the working environment relaxed and entertaining.

The current and past members of the Digital Data Embedding laboratory and the visiting scholars that joined us in our work, also deserve a tremendous amount of my thanks. They provided me with help, encouragement, and friendship.

Further, I wish to acknowledge all my friends who made my stay in Binghamton quite enjoyable. Mainly I would like to thank Arjun Sarath and Pradyumna Kaushik for keeping me sane during the COVID-19 pandemic.

Lastly, my biggest thanks goes to my mother, who always encouraged me to be productive and kept supporting me with any means necessary.

Jan Butora

Binghamton, 2021

Contents

List of tables

List of figures

Notation overview

x ... scalar,
$f = f(x)$ ... function,
$\mathbf{x} = (x_i)$ ... vector,
$\mathbf{x} = (x_{ij})$ ... matrix,
$\mathcal{X}$ ... set,
X ... random variable,
P_X ... distribution of X
$\mathbb{E}[X]$ ... expected value of X
$\mathrm{Var}[X]$ ... variance of X

$\mathbf{X}$ ... cover object,
$\mathbf{Y}$ ... stego object,

$\mathcal{C}$ ... set of all possible covers,
$P^{(c)}$ ... cover object distribution,
$P^{(s)}$ ... stego object distribution,
D_{KL} ... Kullback-Leibler divergence,

β_i ... symmetric change rates,
$\beta_i^{\pm}$ ... assymetric change rates
α ... relative payload,
H_2 ... binary entropy function
H_3 ... ternary entropy function
D ... additive distortion measure
ρ_i ... embedding costs
I_i ... Fisher information
λ ... Lagrange multiplier
δ^2 ... deflection coefficient
$\mathbb{Z}$ ... set of all integers
$\mathbb{R}$ ... set of all real numbers
e_i ... rounding errors

$[x]$ ... Iverson bracket if x is a logical value,
$[x]$ ... rounded value if x is a number,

$|x|$ ... absolute value of x,
$|\mathcal{X}|$ ... cardinality of set $\mathcal{X}$,
Γ ... gamma function

N_1, N_2 ... image height and width

P_{E} ... minimal probability of error
... under equal priors
P_{FA} ... probability of false alarm
P_{MD} ... probability of missed detection
Q_L ... quantizer,

d_{kl} ... unquantized DCT coefficients
q_{kl} ... quantization steps
c_{kl} ... quantized DCT coefficients
DCT ... discrete cosine transformation,
f_{kl}^{ij} ... discrete cosines
i, j ... spatial domain indices
k, l ... DCT domain indices
$\odot$ ... elementwise product
$\oslash$ ... elementwise division
$\mathcal{U}[a, b]$ ... uniform distribution on $[a, b]$
$\mathcal{N}(\mu, \sigma^2)$ ... Gaussian distribution with mean μ
... and variance σ^2
$\triangleq$ ... defining new concept

$\mathbf{X}^{(a,b)}$ ... (a, b) th block of quant. DCT coeff.,

Preface

Steganography and steganalysis are today governed by heuristic rules and formulas. During his four years of the Ph.D. studies at Binghamton University, New York, the author's research focused mainly on bringing some understanding into these heuristics and replacing them with statistically sound viewpoints. This dissertation is written as the final requirement of the author's Ph.D. studies and describes in detail several of the author's findings, all of which were published and peer-reviewed.

In Chapter 1, we introduce the notation and explain the basic concepts, and tools this dissertation builds on. Most importantly, we introduce cost-based and model-based approaches to steganography as well as machine learning and deep learning detectors for steganalysis.

Chapters 2 – 5 focus on improving steganography by using various statistical models within digital images. First, the effect of JPEG quality on the steganographic security is investigated in Chapter 2. Chapter 3 takes our efforts into JPEG images compressed with the so-called "trunc" quantizer, which dramatically affects the security of JPEG steganography. Images compressed with the trunc quantizer naturally lead us to developing a new rule of incorporating side information into the embedding schemes in Chapter 4. In Chapter 5 we show a universal way of converting cost-based steganography into model-based.

Chapters 6 – 7 introduce Reverse JPEG Compatibility Attack, a very powerful and robust attack disabling use of any steganography for high quality JPEG images. We explain the basic assumptions and conditions under which the attack works, as well as its thorough evaluation in Chapter 6. Chapter 7 then focuses on extending the attack into doubly compressed JPEG images.

Using models derived in Chapter 7, we revisit the idea of perturbed quantization in Chapter 8, where we generate side information for a given JPEG image through further compression.

The dissertation is concluded in Chapter 9.

Chapter 1

Introduction and preliminaries

Steganography is another term for covert communication. Instead of communicating the actual message directly, or its encrypted form, it is hidden (embedded) in another cover object, which has the role of a mere decoy. Digital images are especially convenient covers for steganography because their individual elements (pixels or DCT coefficients in a JPEG file) can be slightly modified without changing the semantic meaning of the image. The main requirement for steganography is that the stego objects carrying secrets should be statistically indistinguishable from cover objects, while carrying as much hidden information as possible. Steganalysis on the other hand is the art of detecting steganography. That is typically done by inspecting the statistics of communicated objects and looking for outliers. Once the existence of a steganography can be reliably established, the steganographic system is considered broken even if the adversary cannot read the secrets.

1.1 Steganographic channel

There are three basic approaches for covert communication: steganography by cover modification, cover synthesis, and cover selection [43]. Steganography by cover modification takes existing cover object and modifies it into a stego object, so that it contains the secret message. Steganography by cover synthesis creates a new stego object from scratch, while making it statistically indistinguishable from cover objects. Cover selection simply picks already existing cover that already communicates the secret. There are different drawbacks for each of these techniques. For example, cover selection seems to be the best option, however it is not easy to implement, because searching for a cover object that communicates a secret message in an established way might be computationally infeasable for larger messages. This already violates our assumption on communicating secretely large amount of information. In this work, we only focus on cover modification, as we can take any cover object and introduce few changes to communicate a desired message.

One of the most popular formulations of a steganographic system is the so-called prisoner's problem [118]. There are two prisoners Alice and Bob in separate cells who are allowed to communicate with each other through a communication channel, which is monitored by a warden Eve. Alice and Bob are aware that Eve is eavesdropping on their communication and they want to hatch an escape plan without Eve knowing. However, if Eve suspects that a secret communication takes places, she prohibits all communication. Cryptography in this scenario doesn't help, as Eve would immediately notice that suspicously looking messages are being communicated. To this end, Alice and Bob use steganography to communicate secretely through innocently looking objects.

Eve can be either an active or a passive warden. Active warden can intercept messages between Alice and Bob, modify them in order to destroy potential secrets or can even try to impersonate one of the prisoners and send misleading secret messages. Passive warden can only observe the communication

and collect statistics about the messages, but cannot tamper with them in any way. We will only consider the passive warden scenario in this dissertation.

1.1.1 Formal definition

As already mentioned, steganography uses existing communication channel between two parties Alice and Bob. This channel is potentially eavesdropped by Eve, so Alice and Bob share beforehand a set of secret keys $\mathbf{k} \in \mathcal{K}$. The steganographic system that Alice and Bob can now establish constitutes of a cover source $\{\mathcal{C}, P^{(s)}\}$, a message source $\{\mathcal{M}, P_M\}$, the set of stego keys $\mathcal{K}$, and embedding and extracting functions Emb and Ext. The cover source is made of a set of all possible cover objects $\mathbf{x} \in \mathcal{C}$ and their distrubution $P^{(c)}$. Similarly, the message source is defined by the set of all possible messages $\mathbf{m} \in \mathcal{M}$ and their distribution $P^{(m)}$. Let us now denote $\mathcal{S}$ the set of all possible stego objects with distribution $P^{(s)}$. The embedding function $\text{Emb} : \mathcal{C} \times \mathcal{M} \times \mathcal{K} \to \mathcal{S}$ takes a cover object, a message we want to communicate, a shared secret key and creates a stego object carrying the message $\mathbf{y} = \text{Emb}(\mathbf{x}, \mathbf{m}, \mathbf{k}) \in \mathcal{S}$. Extraction function $\text{Ext} : \mathcal{S} \times \mathcal{K} \to \mathcal{M}$ then takes the stego object and extracts the secret $\mathbf{m} = \text{Ext}(\text{Emb}(\mathbf{x}, \mathbf{m}, \mathbf{k}), \mathbf{k})$ for all $\mathbf{x} \in \mathcal{C}, \mathbf{m} \in \mathcal{M}$, and $\mathbf{k} \in \mathcal{K}$. Steganographic channel we just described is visualized in Figure 1.1.1.

Akin to cryptography, we assume the Kerchoff's principle, which dictates that the warden knows everything about the steganographic channel, apart from the secret key. This includes the cover source, message source as well as embedding and extraction functions. Kerchoff's principle is used as a worst case scenario for the communicating parties, but it prevents security through obscurity, which is in many cases undesirable. It is worth mentioning that steganography is assumed to be repetitive. This means that Alice and Bob keep exchanging messages, giving the warden the ability to collect better statistics about the communicated objects. That is also why Alice and Bob share a set of secret keys. They cannot reuse a secret key during the communication as that would introduce serious security flaws [84, 110].

In this work, we use only two types of cover objects: spatial domain digital images (represented by their pixel values) and JPEG compressed digital images (represented by their DCT coefficients). Regardless of a cover object, we now have some restrictions on the cover elements (pixels, DCT coefficients). For example pixels can only attain values between 0 and 255. Similar conditions can be derived for DCT coefficients as well. We can imagine that, if we aren't careful, the embedding algorithm can create a stego object outside the set of all possible covers $\mathbf{y} \notin \mathcal{C}$. Since these boundary cases can be prevented during the embedding, we will always assume that $\mathcal{C} = \mathcal{S}$ and we will usually be only interested in comparing their distributions $P^{(c)}$ and $P^{(s)}$.

To avoid any biases during the embedding procedure, messages are assumed to be random uncorrelated bits. For this reason, steganography is typically preceded with cryptography, which randomizes the secret message bits. This requires Alice and Bob to share another pair of cryptographic keys before the secret communication is established. Since, we assume the secret messages to be sequences of random bits, throughout the whole dissertation we won't embed any actual secrets, only simulate embedding of random messages. We only need to establish a measure of how much information are we communicating. Standard quantity in literature is the relative payload α, which is measured in bits per pixel (bpp) for spatial domain and bits per non-zero AC DCT coefficients (bpnzac) in the DCT (JPEG) domain.

1.1.2 Formal security

Having the cover source $\{\mathcal{C}, P^{(c)}\}$ we can consider any cover object as realization of a random variable following the cover distribution $\mathbf{X} \sim P^{(c)}$. Similarly, we can say view a stego object as a realization of a random variable following the stego distribution $\mathbf{Y} \sim P^{(s)}$. It is only natural, to say that steganographic system is secure, if the cover and stego distributions are statistically

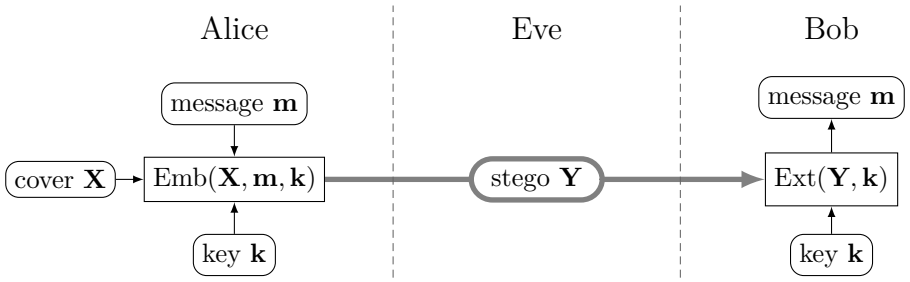

Figure 1.1.1: Steganographic channel.

indistinguishable. To measure the distance between these two probability distributions, we will be using Kullback-Leibler divergence (KL divergence),

$$D_{\mathrm{KL}}(P^{(c)}||P^{(s)}) = \sum_{\mathbf{x}\in\mathcal{C}} P^{(c)}(\mathbf{x}) \log \frac{P^{(c)}(\mathbf{x})}{P^{(s)}(\mathbf{x})}, \tag{1.1.1}$$

which is a fundamental concept from information theory. The stego system is then called perfectly secure (undetectable), if $D_{\mathrm{KL}}(P^{(c)}||P^{(s)}) = 0$, which happens exactly when the distributions are the same $P^{(c)} = P^{(s)}$ making it impossible for Eve to distinguish between the cover and stego objects. While this is the security we would like to have, in practice it is not easy to achieve. The security requirements are thus relaxed a little and we call a stegosystem ϵ-secure, if $D_{\mathrm{KL}}(P^{(c)}||P^{(s)}) \leq \epsilon$.

1.2 Steganography

Early steganographic algorithms using digital images were hiding messages into the Least Significant Bits (LSB) of the image's pixel values. In Figure 1.2.1 we see an image together with its LSB plane (black for zero and white for one). Because of it's noise-like appearance, hiding data by changing LSBs used to be considered secure. The noise pattern in the LSB plane is present due to various noises inside the digital imaging sensors (shot noise, electric noise, etc.). Existence of this noise prevents us from hiding information into computer generated images, because those typically don't contain any noise. The simplest algorithm using the LSB plane is the LSB Replacement (LSBR). The secret stego key is used to create a pseudo-random path across the cover elements (pixels or DCT coefficients) and the message bits are embedded into their LSB values. If the cover LSB doesn't match the message bit, the algorithm simply flips the LSB. This is an example of a non-adaptive embedding scheme, because the embedding changes can be potentially anywhere in the image. In the next section we will discuss content-adaptive steganography. Even though the LSBR keeps the visual properties of a random noise, the LSB plane is not completely random (see the saturated headlights in the LSB plane of Figure 1.2.1) and many powerful attacks exist against this algorithm [47, 37, 79, 46, 88, 27, 39, 144, 124, 125].

More recent version of this embedding algorithm is LSB Matching (LSBM), which matches the cover LSB to a message bit by randomly changing the pixel value by $+1$ or -1, if the LSB doesn't carry the message bit. Even though LSBM can change more than one bit per pixel, it doesn't introduce any characteristic artifacts into the histogram, as was the case for LSBR, which is why LSBM is used for content-adaptive steganography too. However, due to its non-adaptive behavior, there still exist many accurate attacks on this scheme [28, 25, 80, 81, 100].

For non-adaptive schemes, typical measure of distortion between cover and stego images is the hamming distance,

$$d_H(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} [x_i \neq y_i], \tag{1.2.1}$$

where n is the number of pixels in the image and $[\cdot]$ is the Iverson bracket, which equals one if the expression inside is satisfied, otherwise equals zero. Having a measure of distortion between the images, we can compute the change rates, defined as the average distortion per pixel (DCT coefficient),

$$\beta = \frac{1}{n} d_H(\mathbf{x}, \mathbf{y}). \tag{1.2.2}$$

One thing to notice here, is that the change rate is the same for every cover element. We'll see in the next section that using more general distortion metric for content-adaptive schemes will yield different change rates β_i for every cover element.

So far, we didn't consider any coding of the secret message, meaning we could simply extract the message from the LSBs of the stego image. Inspired by coding techniques used in error-correcting codes, same principles can be applied for steganography. Having a binary parity-check matrix $\mathbf{H}$ and a secret message $\mathbf{m}$, we want the column vector of the stego image LSBs $\mathbf{y}$ to satisfy $\mathbf{m} = \mathbf{H}\mathbf{y}$. Depending on the matrix $\mathbf{H}$, there can be several solutions to this problem and we would like to minimize the distortion we introduce to the cover image $\mathbf{x}$. This can be stated as minimizing the hamming distance (1.2.1) between $\mathbf{x}$ and $\mathbf{y}$. The embedding process can then be reformulated as an optimization problem

$$\min d_H(\mathbf{x}, \mathbf{y}) \tag{1.2.3}$$

while communicating the correct message

$$\mathbf{m} = \mathbf{H}\mathbf{y} \tag{1.2.4}$$

This process is typically referred to as matrix embedding [43] and the first algorithm using this embedding method was F5 [128] and its improved version nsF5 [57], both using a Hamming matrix as its parity-check matrix. Introducing coding into the embedding process reduces the change rates (1.2.2) significantly. With development of LSBM, this can be further improved with ternary embedding over binary. Instead of considering LSBs of the cover image, which can be mathematically written as $x_i \mod 2$, we use the values $x_i \mod 3$. This can indeed be done, because an embedding change is a ternary variable achieving values $-1, 0, 1$. There has been some effort in embedding changes of larger magnitude, for example in [116] pentary embedding is used, but changes of cover elements by $+2$ and -2 need to be done very carefully to avoid detectable distortion and might not bring much improvement in practical security.

Latest development in coding for steganography introduces so-called Syndrome Trellis Codes (STCs) [41], which is a parallel to convolutional error-correcting codes, utilizing the Viterbi algorithm for decoding. Given a distortion metric between a cover and stego objects $D(\mathbf{x}, \mathbf{y})$, STCs minimize the distortion while achieving near optimal performance on the Rate-Distortion (RD) bound. The RD bound gives lower bound on the change rate

$$\beta \geq H_3^{-1}(\alpha), \tag{1.2.5}$$

where H_3^{-1} is the inverse of the ternary entropy function

$$H_3(x) = -(1-2x)\log_2(1-2x) - 2x\log_2 x. \tag{1.2.6}$$

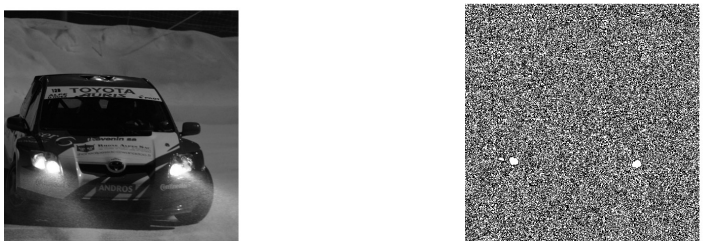

Figure 1.2.1: An image and its pixel LSB plane.

We encourage the reader to read the original paper for more details on STCs.

Due to the near-optimal performance of the STCs, we will always assume in this work the optimal coding given by the RD bound (1.2.5). As a consequence, the condition (1.2.4) will be replaced by the following condition on the changes rates

$$\sum_{i=1}^{n} H_3(\beta_i) = n\alpha. \tag{1.2.7}$$

To avoid a potential confusion later, we want to specify that we denote β_i a one-sided change rate. That is a probability of change by $+1$ and by -1 are both equal to β_i, unless stated otherwise. The probability of introducing no change is then $1-2\beta_i$. This is just a convention, as one can define β_i to be the total change rate. However in such case, we would have to use a slightly modified expression for the ternary entropy (1.2.6).

1.2.1 Content-adaptive algorithms

A huge improvement for steganography came in with content-adaptivity. In particular, we inform the embedding scheme of the content in order to avoid making changes in easy-to-model parts of the image. Intuitively, this makes sense, because making a change, even if only by $+1$ or -1, to a pixel in a neighborhood that is very easy to model, should be highly detectable. This is typically done by assigning embedding costs of changing cover elements. As of today, there exist three main approaches to content-adaptive steganography: cost-based, model-based and adversarial. We will now briefly explain each of these strategies.

1.2.1.1 Cost-based

The first and most popular content-adaptive strategies use heurestically defined costs ρ_i of changing i-th cover element. These costs are typically determined through experimentation and a feedback from current state-of-the-art detectors. The total distortion is then defined as a sum of these costs over all changes (cost of not changing a cover element is zero),

$$D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \rho_i[x_i \neq y_i]. \tag{1.2.8}$$

This distortion implicitly assumes that the embedding changes don't interact with each other, otherwise we wouldn't be able to simply sum individual costs. Still, minimizing expression (1.2.8) with condition (1.2.7) is computationally infeasable (we don't apriori know where the embedding changes will be), thus we instead minimize the expected distortion

$$\mathbb{E}[D(\mathbf{x},\mathbf{y})] = \sum_{i=1}^{n} \rho_i \beta_i. \tag{1.2.9}$$

For simplicity, we will be denoting this distortion as $D(\mathbf{x},\mathbf{y})$. From (1.2.9) we can now compute the optimal change rates minimizing the distortion as

$$\beta_i = \frac{e^{-\lambda\rho_i}}{1+2e^{-\lambda\rho_i}}, \tag{1.2.10}$$

where $\lambda > 0$ is a Lagrange multiplier determined by the payload constraint (1.2.7). We'd like to point out that non-adaptive distortion (1.2.1) can also be written as cost-based distortion (1.2.8), with all costs being equal to the same constant value. Among the most secure cost-based algorithms are HUGO [109], WOW [70], S-UNIWARD [74], and HILL [98] in spatial domain and J-UNIWARD [74], and UERD [67] in JPEG domain.

1.2.1.2 Model-based

Another approach to steganography tries to avoid heuristically defined costs and uses a statistical model instead. Let $p_i^{(c)}(x)$ be the probability distribution of the cover model at i-th element. For steganography that uses LSBM embedding, we can model the stego distribution as

$$p_i^{(s)}(x) = (1-2\beta_i)p_i^{(c)}(x) + \beta_i p_i^{(c)}(x+1) + \beta_i p_i^{(c)}(x-1). \tag{1.2.11}$$

With these models, we can minimize the statistical detectability, which is asymptotically directly linked to the so-called deflection coefficient

$$\delta^2 \propto \sum_{i=1}^{n} \beta_i^2 I_i \tag{1.2.12}$$

where I_i is the steganographic Fisher Information

$$I_i = \int_{\mathbb{R}} \frac{1}{p_i^{(c)}(x)} \left(\frac{\partial p_i^{(s)}(x)}{\partial \beta_i}\bigg|_{\beta_i=0} \right)^2. \tag{1.2.13}$$

In particular, the optimal change rates satisfy for each i

$$\beta_i I_i = \frac{1}{\lambda} H_3^{'}(\beta_i), \tag{1.2.14}$$

where $H_3^{'}(x)$ is the derivative of $H_3(x)$,

$$H_3^{'}(x) = \log_2 \frac{1-2x}{x} \tag{1.2.15}$$

and $\lambda > 0$ is the Lagrange multiplier, subject to the usual payload constraint (1.2.7). In practice this is usually done by solving (1.2.14) and (1.2.7) numerically with a binary search over λ [56, 115, 116].

The first truly model-based algorithms are MiPOD [115] in spatial domain and its recently proposed JPEG version J-MiPOD [24].

1.2.1.3 Adversarial

Adversarial examples attacking deep learning detectors are nothing new in the computer vision field [64, 65, 120, 106]. It is only logical that this approach was successfully utilized for steganography too. Nowadays, the main work dealing with adversarial examples for steganography is ADV-EMB [122]. It divides the cover image elements into two groups: common group for steganographic embedding and adjustable group for adversarial embedding. Given costs from a cost-based stego algorithm, ADV-EMB first embeds a portion of the secret message into the common group and then iteratively modifies the costs of elements belonging to the adjustable group in order to fool the target neural network steganalyzer. In this dissertation, we don't consider the adversarial examples and the reader is therefore encouraged to read the original publication for more information.

1.2.2 Side information

The most secure steganographic schemes are by far the ones utilizing side information. Side information generally comes in form of rounding errors after some information-losing processing of an image, because such processing functions are followed by rounding to integers. Examples of such processing are resizing, JPEG compression, conversion from color to grayscale etc. Because the embedding takes place on the cover image, which is always integer valued, the image before rounding to integers is referred to as the precover. Let $x_i \in \mathbb{R}$ be the precover value of i-th element. The side information is then the rounding error $e_i = x_i - [x_i]$, $-1/2 \leq e_i \leq 1/2$, where $[\cdot]$ is the operation of rounding to the nearest integer. Let us denote $\rho_i(\pm 1)$ the embedding costs of changing the cover element by $+1$ or -1. It was proposed in [31] to modulate the cost of changing $[x_i]$ to $[x_i] + \text{sign}(e_i)$, while keeping the cost in the opposite direction intact,

$$\rho_i(\text{sign}(e_i)) = \rho_i(1 - 2|e_i|) \tag{1.2.16}$$
$$\rho_i(-\text{sign}(e_i)) = \rho_i, \tag{1.2.17}$$

where ρ_i is the original symmetric cost, computed from an embedding algorithm. The embedding finds optimal assymetric change rates by minimizing

$$D(\mathbf{x}, \mathbf{y}) \sum_{i=1}^{n} \rho_i(+1)\beta_i^+ + \rho_i(-1)\beta_i^- \tag{1.2.18}$$

with payload constraint

$$H_3(\beta_i^+, \beta_i^-) = n\alpha \tag{1.2.19}$$

where $H_3(\beta_i^+, \beta_i^-)$ is the ternary entropy function for assymetric change rates

$$H_3(\beta_i^+, \beta_i^-) = -(1 - \beta_i^+ - \beta_i^-)\log_2(1 - \beta_i^+ - \beta_i^-) - \beta_i^+ \log_2 \beta_i^+ - \beta_i^- \log_2 \beta_i^- \tag{1.2.20}$$

The optimal change rates can be computed as

$$\beta_i^{\pm} = \frac{e^{-\lambda\rho_i(\pm 1)}}{1 + e^{-\lambda\rho_i(+1)} + e^{-\lambda\rho_i(-1)}}. \tag{1.2.21}$$

where it is assumed that β_i^+ and β_i^- are independent of each other.

Previously, we only considered symmetric embedding costs $\rho_i = \rho_i(+1) = \rho_i(-1)$, but now we can see that the side-information destroys this symmetry. As a result, the average change rate is typically higher than for non-informed schemes, because ternary entropy (1.2.20) achieves its maximum for $\beta_i^+ = \beta_i^-$.

1.3 Steganalysis

Steganalysis is the practice of detecting steganography. Unlike cryptanalysis, the steganalysis doesn't have to necessarily extract the secret message, because such task is usually unattainable. The goal of steganalysis is instead to only establish the presence of the secret communication with some non-trivial probability. A specialized discipline called forensic steganalysis is devoted to extracting the secret message [43].

The most natural way of performing steganalysis can be formulated as a simple hypothesis testing

$$\mathrm{H}_0 : \mathbf{x} \sim P_c, \tag{1.3.1}$$
$$\mathrm{H}_1 : \mathbf{x} \sim P_s. \tag{1.3.2}$$

It can be shown that under the Neyman-Pearson setting, the optimal detector can be derived for this problem using the Likelihood-Ratio Test (LRT). In practice however, such test is not easy to implement, because we either don't have good estimates of the distributions P_c, P_s or the dimensionality of the image representation is too big, which makes the problem computationally infeasible. This doesn't mean that steganalytic attacks in form of hypothesis testing do not exist. In fact, there were several of them proposed, ranging from chi-square attack [129] on J-steg [126], histogram attack [49] on OutGuess [111], to Sample pair analysis [101, 38, 78, 82, 83] against embedding algorithms based on LSBR. All these attacks have one thing in common, they are used against non-adaptive algorithms.

Statistical attacks against content-adaptive schemes are not easy to carry out, because nor the images nor the steganography are easily modelable. Nevertheless, we did discover a nice simple attack during the ALASKA [22] steganographic competition. We noticed that the embedding scripts for even modern scheme such as J-UNIWARD [74] prevent the embedding from making changes to DCT coefficients c_i, for which $|c_i| > 1023$. However the DCT coefficients are bounded by smaller values depending on the DCT mode and the JPEG quality factor as we pointed out in [137]. The embedding can thus potentially introduce a DCT coefficient that is not attainable for any cover image. We can then test for these out-of-range (OOR). Such test isn't by itself very powerful, but whenever there is an OOR coefficient in an image, we are 100% confident that it is a stego image, while not introducing any false alarms (cover images detected as stego).

Because the huge dimensionality of images prevents us from using statistical tests directly, steganalysis is in the last few decades mainly performed with machine learning (ML) tools. For the last five years, this paradigm of detection steganography is slowly transitioning to steganalysis with deep learning (DL) detectors. Both these approaches, ML and DL, are generally trained in a supervised fashion. Testing a specific image thus first requires the detector to be trained on images coming from the same image source. The so-called cover-source mismatch (CSM) happens when different source of cover images is used for training and testing. It was shown that CSM can very badly influence the detectors' performance [61, 92, 90, 102], especially for DL detectors [17, 141]. Following the Kerchoff's principle, we will always assume that the cover source is known.

In this work, we will measure performance of the detectors with the minimum probability of error under equal priors. It is the most popular choice of measuring steganographic security and is defined as

$$P_{\mathrm{E}} = \frac{1}{2} \min_{P_{\mathrm{FA}}} (P_{\mathrm{FA}} + P_{\mathrm{MD}}(P_{\mathrm{FA}})), \tag{1.3.3}$$

where P_{FA} stands for probability of false alarm - detecting a cover image as stego - and P_{MD} stands for probability of missed detection - detecting a stego image as cover.

1.3.1 Machine learning

To get around the huge dimensionality problem, Eve might want to represent the image in terms of some heurestically determined features computed from the image. These features then inform her of the properties of the image, such as histogram, higher order co-ocurrences etc., hoping that steganography changes these properties enough to allow for detection. Having collected the features from cover and stego images across the whole training set, she can then train a ML classifier that will try to classify images into stego and cover classes based on their feature representation. The most common features sets for steganalysis, typically called Rich Models (RM), are JRM [96], DCTR [72], and GFR [119] in JPEG domain and SRM [54] in spatial domain. If Eve knows what kind of steganographic algorithm is being used, she can compute the Selection Channel - change rates β_i - for every image and use it to generate more reliable feature sets. Examples of these Selection Channel Aware feature sets are SCA-GFR, SCA-DCTR [30], and maxSRM [35].

Because the dimensionality of the feature sets we mentioned above is still in order of tens of thousands, training a standard ML classifier, such as Support Vector Machines (SVM), would require a lot of training data to prevent overfitting. Obtaining this many images might not be feasible in reasonable time and even if it was, training SVMs with so many high-dimensional features would be extremely computationally demanding. An alternative random forest classifier was thus proposed in form of Ensemble Classifer (EC) [97], ensembling several Fisher Linear Discriminant (FLD) base learners. EC is much easier to train than SVM due to its simplicity, while providing comparable detection. In [26] it was shown that the ensemble classifier behaves as regularized linear classifier and proposed even faster alternative, the Low-Complexity Linear Classifier (LCLC).

1.3.2 Deep learning

With the recent boom in deep learning over the past decade, Convolutional Neural Networks (CNN) found their application in steganalysis too. Unlike feature based steganalysis, using a CNN doesn't require the steganalyst to choose or design appropriate feature set, as the network is trained in an end-to-end fashion without any hand-crafted features. The main and only work is the network design. Moreover, because of the complex nature of the neural networks, their training needs to be performed on specialized GPUs.

The first CNN successfully used for steganalysis was the XuNet [131]. It was believed that CNNs in steganalysis need to be told how the first layer convolutional filters should be initialized to better extract the specific noise patterns introduced by steganography, so XuNet used fixed high-pass filters in the first convolutional layer as part of image preprocessing during training and testing. The next success was the YeNet [133], which improved the detection power drastically over XuNet. This network also had manually defined filters in the first layer with SRM filters. Both YeNet and XuNet were designed only for spatial domain steganalysis. The SRNet [8] was the first network without any domain specific implementation tricks while achieving superior detection in both spatial and JPEG domain. Design of SRNet was however influenced by a domain specific belief that pooling operation used early in the network cripples the steganalysis performance, and thus pooling was avoided in the first seven convolutional layers. Recently, during the ALASKA 2 [138, 20, 23] steganalysis competition, it has been observed that using CNNs without any steganography specific elements, designed for computer vision tasks, such as EfficientNet [105] family, can be applied to JPEG steganalysis with state-of-the-art performance. It was later shown that these off-the-shelf CNNs can be further improved for steganalysis by avoiding pooling and striding in the early layers [136].

1.4 JPEG compression

JPEG compression proceeds by dividing the image into 8×8 blocks, applying the DCT to each block, dividing the DCT coefficients by quantization steps, and rounding to integers. The coefficients are

then arranged in a zig-zag fashion and losslessly compressed to be written as a bitstream into the JPEG file together with a header. We first describe this process for a grayscale image.

For better readability, everywhere in this work, i, j will be strictly used to index pixels and k, l will index DCT coefficients. The original uncompressed 8-bit grayscale image with $N_1 \times N_2$ pixels is denoted $\mathbf{x} \in \{0, 1, \dots, 255\}^{N_1 \times N_2}$. For simplicity we assume that N_1 and N_2 are multiples of 8. Constraining $\mathbf{x} = (x_{ij})$ to one specific 8×8 block, we will use indices $0 \le i, j \le 7$ to index the pixels in this block. During JPEG compression, the DCT coefficients before quantization, $d_{kl} \in \mathbb{R}$, are obtained using the formula $d_{kl} = \mathrm{DCT}_{kl}(\mathbf{x}) \triangleq \sum_{i,j=0}^{7} f_{kl}^{ij} x_{ij}$, $0 \le k, l \le 7$, where

$$f_{kl}^{ij} = \frac{w_k w_l}{4} \cos \frac{\pi k(2i+1)}{16} \cos \frac{\pi l(2j+1)}{16}, \tag{1.4.1}$$

$w_0 = 1/\sqrt{2}$, $w_k = 1$ for $0 < k \le 7$ are the discrete cosines. Before applying the DCT, each pixel is adjusted by subtracting 128 from it during JPEG compression, a step we omit here since it has no effect on our analysis.

The quantized DCTs are $c_{kl} = [d_{kl}/q_{kl}]$, $c_{kl} \in \{-1024, \dots, 1023\}$, where q_{kl} are quantization steps in a luminance quantization matrix, which is supplied in the header of the JPEG file.

Denoting the 8×8 matrix of ones with boldface $\mathbf{1}$, the standard quantization matrix for quality factor $Q \in \{1, 2, \dots, 100\}$ is

$$\mathbf{q}(Q) = \begin{cases} \max\left\{\mathbf{1}, \left[2\mathbf{q}(50)\left(1 - \frac{Q}{100}\right)\right]\right\}, & Q > 50 \\ \min\left\{255 \times \mathbf{1}, \left[\mathbf{q}(50)\frac{50}{Q}\right]\right\}, & Q \le 50, \end{cases} \tag{1.4.2}$$

where the luminance quantization matrix for quality factor 50 is

$$\mathbf{q}(50) = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}. \tag{1.4.3}$$

During decompression, the above steps are reversed. For a block of quantized DCT coefficients c_{kl}, the corresponding block of non-rounded pixel values after decompression is $y_{ij} = \mathrm{DCT}_{ij}^{-1}(\mathbf{c} \cdot \mathbf{q}) \triangleq \sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl} c_{kl}$, $y_{ij} \in \mathbb{R}$. To obtain the final decompressed image, y_{ij} are rounded to integers and clipped to a finite dynamic range $[0, 255]$.

For compression of color images, the RGB representation is typically changed to YC_bC_r (luminance, and two chrominance signals) with:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ C_b &= 128 - 0.169R - 0.331G + 0.5B \\ C_r &= 128 + 0.5R - 0.419G - 0.081B. \end{aligned} \tag{1.4.4}$$

The luminance Y is processed as above, while the chrominance signals are optionally subsampled, then transformed using DCT, and finally quantized with chrominance quantization matrices, also stored in the header of the JPEG file. For the chrominance quantization table $\mathbf{q}_C(Q)$ at quality Q, the same formula (1.4.2) applies with chrominance quantization table at quality 50

$$\mathbf{q}_C(50) = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}. \tag{1.4.5}$$

In this work we avoid subsampling of chrominance signals because its effect on steganography has not been thoroughly studied yet. For more detailed description of the JPEG format, the reader is referred to [108].

Chapter 2

Effect of JPEG Quality on Steganographic Security

This work investigates both theoretically and experimentally the security of JPEG steganography as a function of the quality factor. For a fixed relative payload, modern embedding schemes, such as J-UNIWARD and UED-JC, exhibit surprising non-monotone trends due to rounding and clipping of quantization steps. Their security generally increases with increasing quality factor but starts decreasing for qualities above 95. In contrast, old-fashion steganography, such as Jsteg, OutGuess, and model-based steganography, exhibit complementary trends. The results of empirical detectors closely match the trends exhibited by the KL divergence computed between models of cover and stego DCT modes. In particular, our analysis shows that the main reason for the complementary trends is the way modern schemes attenuate embedding change rates with increasing spatial frequency. Our model also provides guidance on how to adjust the embedding algorithm J-UNIWARD to improve its security for JPEG quality factor 100.

2.1 Introduction

The JPEG format is the most ubiquitous image format in use today due to its ability to efficiently compress visual data without introducing perceivable artifacts and the fact that it is supported across all platforms by all applications capable of displaying imagery. It is also a quite complex format because the compression algorithm is controlled by numerous parameters and settings, such as the selection of the color representation, quantization matrices, chrominance subsampling, and the specific implementation of the Discrete Cosine Transform (DCT). Surprisingly little research is available on the effect of the above choices on detectability of steganography.

Arguably, the most influential settings in JPEG compression are the quantization matrices, which control the trade-off between the file size and image quality. As this paper shows using both empirical detectors and theoretical arguments, the impact of quantization on security is quite complex and depends on the specific embedding algorithm. Most notably, for relative payload fixed in terms of bits per non-zero AC DCT coefficient (bpnzac) the security of "old" embedding methods, such as Jsteg [126] (or any generic LSB flipper), OutGuess [111], and Model-Based Steganography (MBS) [114], decreases with increasing JPEG quality factor (QF) but starts increasing for qualities close to 100, while the trend is just the *opposite* for modern embedding schemes, such as J-UNIWARD [74] and UED-JC [67]. Hints of this can be observed, but are not explicitly commented upon, in previous work with steganalyzers implemented using the JPEG Rich Model (JRM) and the JPEG Projection Spatial Rich Model (JPSRM) (Table 1 in [71]), detectors using the JPEG-phase-aware features (Fig. 5 and 6 in [72]), as well as detectors implemented as Convolutional Neural Networks (CNNs) [18].

In Section 2.2, we introduce the notation and the model of DCT coefficients used in Section 2.3 to quantify the impact of embedding using the KL divergence between cover and stego models of individual DCT modes. The datasets used in this paper as well as the estimation of the model from images are described in Section 2.4. Theoretical predictions derived from the model are validated experimentally using machine-learning based steganalyzers in Section 6.5. In Section 6.3, we provide a more intuitive explanation of the observed non-monotone security trends and identify the modulation of change rates across spatial frequencies as the key element responsible for the observed complementary trends. In the same section, we also use our model to find an adjustment of embedding change rates of J-UNIWARD to improve its security for quality factor 100. A summary and future directions appear in Section 7.5.

2.2 JPEG image model

In this section, we first introduce the notation followed by a model of JPEG DCT coefficients that will later be used in Section 2.3 to assess the impact of steganographic embedding changes on security.

2.2.1 Notation

For simplicity, we only consider $n_1 \times n_2$ 8-bit grayscale images x_{ij}, $1 \le i \le n_1$, $1 \le j \le n_2$, with n_1 and n_2 multiples of 8. The (a,b)th 8×8 block of pixels, $1 \le a \le n_1/8$, $1 \le b \le n_2/8$, formed by pixels with indices $8(a-1)+i+1$, $8(b-1)+j+1$, $0 \le i,j \le 7$, will be denoted $\mathbf{x}^{(a,b)} = (x_{ij}^{(a,b)})$. Similarly, the (a,b)th 8×8 block of unquantized and quantized DCT coefficients will be denoted $\mathbf{c}^{(a,b)} = (c_{ij}^{(a,b)})$ and $\mathbf{d}^{(a,b)} = (d_{ij}^{(a,b)})$, respectively, where $d_{kl}^{(a,b)} = \left[c_{kl}^{(a,b)}/q_{kl}\right]$ with q_{kl} denoting the luminance quantization steps and $[x]$ the operation of rounding to integers.

Denoting the 8×8 matrix of ones with boldface $\mathbf{1}$, the standard quantization matrix for quality factor $Q \in \{1, 2, \dots, 100\}$ is

$$\mathbf{q}(Q) = \begin{cases} \max\left\{\mathbf{1}, \left[2\mathbf{q}(50)\left(1 - \frac{Q}{100}\right)\right]\right\}, & Q > 50 \\ \min\left\{255 \times \mathbf{1}, \left[\mathbf{q}(50)\frac{50}{Q}\right]\right\}, & Q \le 50, \end{cases} \tag{2.2.1}$$

where the luminance quantization matrix for quality factor 50 is

$$\mathbf{q}(50) = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}. \tag{2.2.2}$$

We use the symbol $\mathbb{Z}$ for the set of all integers, $\Gamma(x)$ for the gamma function, and $H_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$, $H_3(x) = H_2(x) + x$ for the binary and ternary entropy expressed in bits.

2.2.2 Model

Unquantized DCT coefficients $c_{kl}^{(a,b)}$ are modeled as 64 independent channels (modes (k,l)). The coefficients in each mode (k,l) across all blocks in the image (a,b) are assumed to be independent

realizations of a random variable with the generalized Gaussian (GG) distribution

$$c_{kl}^{(a,b)} \sim g(x; \gamma_{kl}, w_{kl}), \tag{2.2.3}$$

with zero mean, shape parameter $\gamma_{kl} > 0$, and width parameter $w_{kl} > 0$:

$$g(x; \gamma, w) = \frac{\gamma}{2w\Gamma\left(\frac{1}{\gamma}\right)} \exp\left(-\left|\frac{x}{w}\right|^{\gamma}\right). \tag{2.2.4}$$

We note that the variance of the GG distribution is $v = \frac{w^2\Gamma(3/\gamma)}{\Gamma(1/\gamma)}$.

Quantized DCTs from a cover image, $d_{kl}^{(a,b)}$, across all blocks (a, b), follow the quantized GG probability mass function $P_{kl}^{(c)}(m) \triangleq \Pr\{d_{kl}^{(a,b)} = m\}$, $m \in \mathbb{Z}$:

$$P_{kl}^{(c)}(m) = \int_{q_{kl}(m-\frac{1}{2})}^{q_{kl}(m+\frac{1}{2})} g(x; \gamma_{kl}, w_{kl})\mathrm{d}x = \omega(m; q_{kl}, \gamma_{kl}, w_{kl}) \tag{2.2.5}$$

$$\omega(m; q, \gamma, w) = \begin{cases} \frac{1}{2}\left[\underline{\Gamma}\left(\frac{1}{\gamma}, \left(\frac{q\left(|m|+\frac{1}{2}\right)}{w}\right)^{\gamma}\right)\right. \\ \left.-\underline{\Gamma}\left(\frac{1}{\gamma}, \left(\frac{q\left(|m|-\frac{1}{2}\right)}{w}\right)^{\gamma}\right)\right] & \text{for } m \neq 0 \\ \underline{\Gamma}\left(\frac{1}{\gamma}, \left(\frac{q}{2w}\right)^{\gamma}\right) & \text{for } m = 0 \end{cases} \tag{2.2.6}$$

where

$$\underline{\Gamma}(x, z) = \frac{1}{\Gamma(x)} \int_0^z t^{x-1} e^{-t} \mathrm{d}t, \tag{2.2.7}$$

is the normalized lower incomplete gamma function.

2.3 Embedding models

For old steganographic systems, it is easier to obtain the impact of embedding on the distribution of quantized DCT coefficients because the schemes are not adaptive to content. Instead, a fixed embedding operation is typically applied to a selected subset of coefficients with a fixed change rate β determined by the size of the secret payload to be embedded.

Using the GG model of cover DCT coefficients, we can express the total expected number of non-zero quantized DCT coefficients N_0, the number of DCT coefficients different from 0 and 1, N_{01}, and the number of non-zero AC DCT coefficients, $N_{0\mathrm{AC}}$, as

$$N_0 = n_1 n_2 \left(1 - \frac{1}{64} \sum_{k,l=0}^{7} P_{kl}^{(c)}(0)\right) \tag{2.3.1}$$

$$N_{01} = n_1 n_2 \left(1 - \frac{1}{64} \sum_{k,l=0}^{7} \left[P_{kl}^{(c)}(0) + P_{kl}^{(c)}(1)\right]\right) \tag{2.3.2}$$

$$N_{0\mathrm{AC}} = n_1 n_2 \left(1 - \frac{1}{64} - \frac{1}{64} \sum_{(k,l)\neq(0,0)} P_{kl}^{(c)}(0)\right). \tag{2.3.3}$$

2.3.1 Generic LSB flipper

By a generic LSB flipper (LSBF), we understand an algorithm that embeds messages by replacing the Least Significant Bits (LSBs) of pseudo-randomly selected quantized DCT coefficients that are not equal to 0 or 1 with message bits. For example, the embedding algorithm Jsteg falls into this category. LSB replacement is the most popular type of steganography because it is simple and can be applied to virtually any sampled signal. As of October 2017, out of 2863 tools available on the Internet capable of hiding data in digital images, 1024 (36%) of them embed secrets by manipulating LSBs.[1]

Assuming an absolute payload of M bits to be embedded, the probability of changing a quantized DCT coefficient not equal to zero or one is thus $\beta = M/(2N_{01})$, where N_{01} is the number of all DCT coefficients in the cover image not equal to zero or one, the maximum number of bits that can be embedded. In terms of the relative payload α in bits per non-zero AC DCT coefficient (bpnzac) and in terms of bits per pixel (bpp), $M = \alpha N_{0\mathrm{AC}}$ and $M = \alpha n_1 n_2$, respectively. Thus, using (2.3.2) and (2.3.3), the change rates w.r.t. N_{01} are

$$\beta = \frac{\alpha N_{0\mathrm{AC}}}{2N_{01}} \quad \alpha \text{ in bpnzac} \tag{2.3.4}$$

$$\beta = \frac{\alpha n_1 n_2}{2N_{01}} \quad \alpha \text{ in bpp.} \tag{2.3.5}$$

Quantized DCT coefficients in the stego image follow the p.m.f. $P_{kl}^{(s)}$, $0 \le k, l \le 7$:

$$\begin{aligned} P_{kl}^{(s)}(2m) &= (1-\beta)P_{kl}^{(c)}(2m) + \beta P_{kl}^{(c)}(2m+1) \quad m \neq 0 \\ P_{kl}^{(s)}(2m+1) &= \beta P_{kl}^{(c)}(2m) + (1-\beta)P_{kl}^{(c)}(2m+1) \quad m \neq 0 \\ P_{kl}^{(s)}(m) &= P_{kl}^{(c)}(m), \quad m \in \{0,1\}. \end{aligned} \tag{2.3.6}$$

2.3.2 OutGuess

OutGuess embedding proceeds in two stages – embedding and correction. First, the secret message is embedded using LSBR as in the generic LSBF. Then, more changes are introduced in unused DCT coefficients to preserve the global histogram of DCT coefficients. This introduces the following impact on quantized DCT coefficients in the stego image:

$$P_{kl}^{(s)}(2m) = \begin{cases} (1-\beta)P_{kl}^{(c)}(2m) + \beta\frac{P^{(c)}(2m)}{P^{(c)}(2m+1)}P_{kl}^{(c)}(2m+1) & m > 0 \\ (1-\beta)P_{kl}^{(c)}(2m) + \beta\frac{P^{(c)}(2m+1)}{P^{(c)}(2m)}P_{kl}^{(c)}(2m+1) & m < 0 \end{cases}$$

$$P_{kl}^{(s)}(2m+1) = \begin{cases} \beta P_{kl}^{(c)}(2m) + (1-\beta)\frac{P^{(c)}(2m)}{P^{(c)}(2m+1)}P_{kl}^{(c)}(2m+1) & m > 0 \\ \beta P_{kl}^{(c)}(2m) + (1-\beta)\frac{P^{(c)}(2m+1)}{P^{(c)}(2m)}P_{kl}^{(c)}(2m+1) & m < 0 \end{cases}$$

$$P_{kl}^{(s)}(m) = P_{kl}^{(c)}(m), \quad m \in \{0,1\}$$

where $P^{(c)}$stands for the global p.m.f. of DCT coefficients in the cover image.

2.3.3 nsF5

For nsF5, the maximum number of bits that can be embedded is equal to the number of non-zero AC DCT coefficients in the cover image, $N_{0\mathrm{AC}}$. Assuming optimal source coding, nsF5 modifies the

[1] N. Johnson, "IoT Forensic Considerations and Steganography Beyond Images." Invited talk presented at the Network and Cloud Forensics Workshop, IEEE Conference on Communications and Network Security, October 9–11, 2017, Las Vegas, Nevada, USA.

$k\backslash l$	0	1	2	3	4	5	6	7
0	2.24	0.43	0.40	0.38	0.37	0.37	0.36	0.35
1	0.48	0.46	0.43	0.43	0.42	0.42	0.41	0.40
2	0.45	0.45	0.44	0.42	0.42	0.42	0.41	0.41
3	0.45	0.45	0.43	0.43	0.42	0.42	0.42	0.41
4	0.44	0.45	0.44	0.42	0.43	0.42	0.42	0.41
5	0.43	0.45	0.44	0.43	0.43	0.42	0.42	0.41
6	0.41	0.44	0.43	0.43	0.42	0.43	0.42	0.42
7	0.40	0.42	0.42	0.42	0.42	0.42	0.42	0.41

$k\backslash l$	0	1	2	3	4	5	6	7
0	709	2.89	1.06	0.52	0.32	0.21	0.14	0.08
1	5.87	2.24	1.08	0.68	0.49	0.34	0.25	0.15
2	2.27	1.47	0.89	0.53	0.39	0.29	0.21	0.15
3	1.46	1.05	0.67	0.49	0.36	0.27	0.19	0.14
4	0.91	0.76	0.57	0.39	0.31	0.24	0.18	0.12
5	0.61	0.57	0.45	0.33	0.27	0.20	0.16	0.11
6	0.36	0.42	0.32	0.27	0.21	0.17	0.13	0.10
7	0.22	0.25	0.22	0.19	0.16	0.13	0.10	0.08

Table 2.1: Top/bottom: Shape/width parameter of GG models of unquantized DCT coefficients in each DCT mode (k, l) estimated from 2000 randomly selected BOSSbase images.

$k\backslash l$	0	1	2	3	4	5	6	7
0	.16807	.26092	.23824	.07496	.05008	.00831	.00485	.00395
1	.26807	.22638	.20590	.05708	.01411	.00121	.00159	.00252
2	.24810	.20875	.07469	.04893	.00455	.00088	.00063	.00196
3	.20128	.06112	.05147	.01152	.00102	.00006	.00018	.00119
4	.05848	.04286	.00513	.00048	.00011	.00001	.00004	.00025
5	.05434	.00646	.00105	.00051	.00006	.00002	.00003	.00021
6	.00630	.00202	.00044	.00012	.00004	.00002	.00005	.00030
7	.00311	.00067	.00030	.00015	.00006	.00014	.00032	.00069

Table 2.2: Average change rates $\overline{\beta}_{kl}$ across DCT modes (k, l) for J-UNIWARD at 0.4 bpnzac for JPEG QF 95 in BOSSbase.

fraction $\beta = H_2^{-1}(M/N_{0\text{AC}})$ of all non-zero AC DCT coefficients, where H_2^{-1} is the inverse binary entropy function. For relative payload α,

$$\beta = H_2^{-1}\left(\frac{\alpha N_{0\text{AC}}}{N_{0\text{AC}}}\right) = H_2^{-1}(\alpha) \quad \alpha \text{ in bpnzac} \tag{2.3.7}$$

$$\beta = H_2^{-1}\left(\frac{\alpha n_1 n_2}{N_{0\text{AC}}}\right) \quad \alpha \text{ in bpp.} \tag{2.3.8}$$

Quantized DCT coefficients in the stego image follow

$$\text{For } (k,l) \neq (0,0): \tag{2.3.9}$$

$$P_{kl}^{(s)}(m) = \begin{cases} (1-\beta)P_{kl}^{(c)}(m) + \beta P_{kl}^{(c)}(m+1) & m > 0 \\ p_{kl}(0) + \beta P_{kl}^{(c)}(1) + \beta P_{kl}^{(c)}(-1) & m = 0 \\ (1-\beta)P_{kl}^{(c)}(m) + \beta P_{kl}^{(c)}(m-1) & m < 0 \end{cases} \tag{2.3.10}$$

$$P_{00}^{(s)}(m) = P_{00}^{(c)}(m).$$

2.3.4 LSBM

We also work out the impact for a generic embedding scheme that uses LSB matching (LSBM) applied to all non-zero DCT coefficients. Even though such an embedding scheme has not been proposed before, it does make sense to include this case in our study for completeness. Denoting the number of all non-zero DCT coefficients with N_0, under optimal source coding the total change rate applied to each non-zero DCT is $\beta = H_3^{-1}(M/N_0)$, where H_3^{-1} is the inverse ternary entropy. For relative payload α,

$$\beta = H_3^{-1}\left(\frac{\alpha N_{0\mathrm{AC}}}{N_0}\right) \quad \alpha \text{ in bpnzac} \tag{2.3.11}$$

$$\beta = H_3^{-1}\left(\frac{\alpha n_1 n_2}{N_0}\right) \quad \alpha \text{ in bpp.} \tag{2.3.12}$$

The stego p.m.f. of quantized DCT coefficients is for $|m| > 1$, $|m| = 1$, and $m = 0$, respectively, and for all k, l :

$$P_{kl}^{(s)}(m) = \begin{cases} (1-\beta)P_{kl}^{(c)}(m) + \frac{\beta}{2}P_{kl}^{(c)}(m+1) + \frac{\beta}{2}P_{kl}^{(c)}(m-1) \\ (1-\beta)P_{kl}^{(c)}(m) + \frac{\beta}{2}P_{kl}^{(c)}\left(m + \frac{m}{|m|}\right) \\ P_{kl}^{(c)}(0) + \frac{\beta}{2}P_{kl}^{(c)}(1) + \frac{\beta}{2}P_{kl}^{(c)}(-1) \end{cases} \tag{2.3.13}$$

2.3.5 J-UNIWARD

The steganographic scheme J-UNIWARD modifies quantized DCT coefficients with probabilities determined by the local content of the cover image. This non-stationarity significantly complicates modeling the impact of embedding. For simplicity, we will assume that J-UNIWARD applies a certain change rate $\overline{\beta}_{kl}$ to all coefficients (including zeros and the DC term) from mode (k, l) in all blocks. These change rates will be determined by averaging the change rates in each DCT mode across a number of images for each JPEG quality factor Q and payload α separately (Section 2.4.2). The impact on the p.m.f. of each DCT mode will thus be for all k, l, m :

$$P_{kl}^{(s)}(m) = (1-\overline{\beta}_{kl})P_{kl}^{(c)}(m) + \frac{\overline{\beta}_{kl}}{2}P_{kl}^{(c)}(m+1) \tag{2.3.14}$$

$$+ \frac{\overline{\beta}_{kl}}{2}P_{kl}^{(c)}(m-1). \tag{2.3.15}$$

Allowing the change rate to be different across the modes captures the fact that the cost of an embedding change in J-UNIWARD depends on the quantization step q_{kl} and thus on the DCT mode. This model is limited, however, because it does not capture the content adaptivity of J-UNIWARD.

2.3.6 UED-JC

In UED steganography (Uniform Embedding Distortion), the cost of changing a DCT coefficient is proportional to its reciprocal value (UED-SC algorithm as originally introduced in [66]). The more advanced version called UED-JC [67] considers four intra and inter-block neighbors of the coefficient to determine the cost (see Section III-C in [67]). This makes the embedding adaptive to content.

To model the impact of embedding, we adopt the same simplification as for J-UNIWARD – the change rates are assumed to depend on the spatial frequency k, l but not on the physical location within the image as in Eq. (2.3.14), and are estimated from a set of images for each quality factor as explained in the next section.

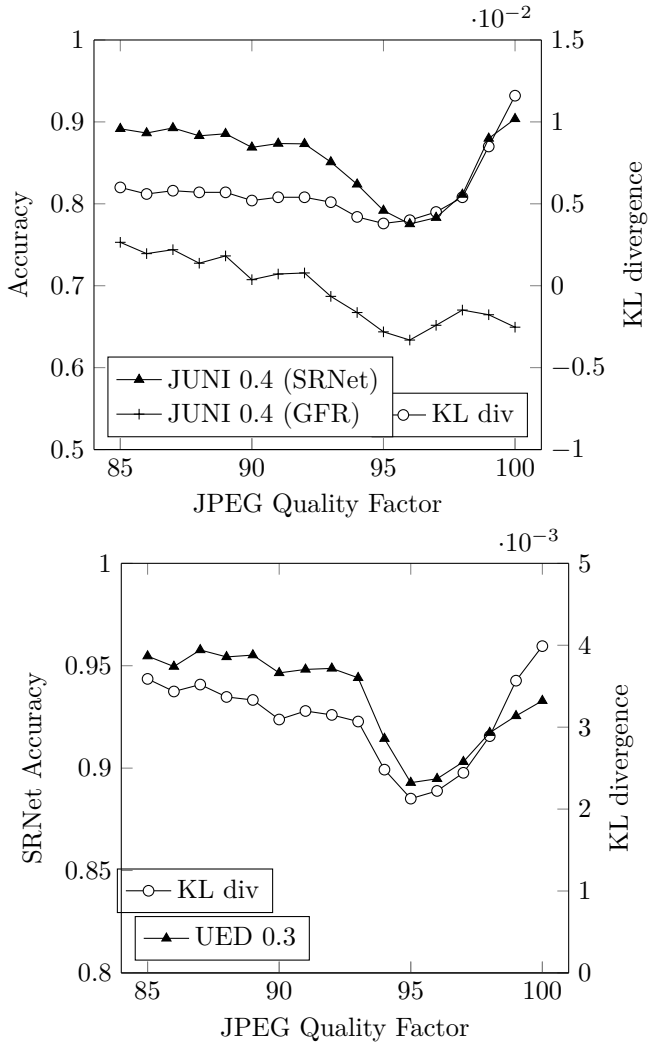

Figure 2.3.1: Left: Detection accuracy of J-UNIWARD as a function of quality factor for payload 0.4 bpnzac using SRNet and GFR with ensemble (left axis) and the KL divergence between cover and stego models for the same payload (right axis). Right: UED-JC for 0.3 bpnzac.

2.3.7 Security

Security will be measured with the KL divergence between the cover and stego p.m.f.s :

$$D_{\mathrm{KL}}(P^{(c)}||P^{(s)}) \triangleq \sum_{k,l=0}^{7} D_{\mathrm{KL}}(P_{kl}^{(c)}||P_{kl}^{(s)}) \tag{2.3.16}$$

$$= \sum_{k,l=0}^{7} \sum_{m=-L}^{L} P_{kl}^{(c)}(m) \log \frac{P_{kl}^{(c)}(m)}{P_{kl}^{(s)}(m)}, \tag{2.3.17}$$

where, for numerical evaluation, L was selected to obtain $P_{kl}^{(c)}(m) < 10^{-15}$ for $|m| > L$.

2.4 Datasets and model estimation

All experiments in this paper were carried out on the union of BOSSbase 1.01 and BOWS2 datasets, each with 10,000 grayscale images, resized from their original size 512×512 to 256×256 using `imresize` with default setting in Matlab. Cover JPEG images were obtained in Matlab using the command `imwrite`. The decompression to the spatial domain for experiments with empirical detectors was obtained by multiplying the DCT coefficients by quantization steps and applying the block inverse DCT without rounding or clipping, `idct2`, in Matlab.

For training empirical detectors, we randomly selected 4,000 images from BOSSbase and the entire BOWS2 dataset with 1,000 BOSSbase images set aside for validation. The remaining 5,000 BOSSbase images were used for testing. In summary, $2 \times 14,000$ cover and stego images were used for training, $2 \times 1,000$ for validation, and $2 \times 5,000$ for testing. This dataset and the split into training and testing has been used for design of many modern deep learning architectures for steganalysis, including the YeNet [133], the Yedroudj-Net [135], and the SRNet [8].

2.4.1 Estimating GG models of DCT modes

A total of $N = 2000$ grayscale uncompressed images were selected from BOSSbase at random and subjected to block-wise DCT without quantization or rounding. The GG parameters shown in Table 2.1 were estimated from all N images using the method of moments [104] for each DCT mode (k, l) separately. Note that the DC term was approximated with a rather wide distribution similar to a Gaussian ($\gamma = 2.24$) while all AC modes exhibit spiky distributions with a similar value of the shape parameter, $0.35 \leq \gamma \leq 0.48$, with the vast majority around $\gamma \approx 0.42$ but a widely varying width $0.08 \leq w \leq 5.87$.

2.4.2 Estimating change rates for J-UNIWARD and UED-JC

Different N randomly chosen images were used for computing the average change rates $\overline{\beta}_{kl}(\alpha, Q)$ for each DCT mode (k, l), payload α, and quality factor Q. Let $\beta_{kl}^{(a,b)}(\mathbf{x}, \alpha, Q)$ denote the change rates returned by the embedding simulator for (a, b)th block in image $\mathbf{x}$. The values $\overline{\beta}_{kl}$ were obtained as averages over all blocks (a, b) and all N images $\mathbf{x}$:

$$\overline{\beta}_{kl}(\alpha, Q) = \frac{64}{N n_1 n_2} \sum_{a=1}^{n_1/8} \sum_{b=1}^{n_2/8} \sum_{\mathbf{x}} \beta_{kl}^{(a,b)}(\mathbf{x}, \alpha, Q). \tag{2.4.1}$$

For compactness, in the rest of this paper we will often drop the explicit dependence of $\overline{\beta}_{kl}$ on α and Q.

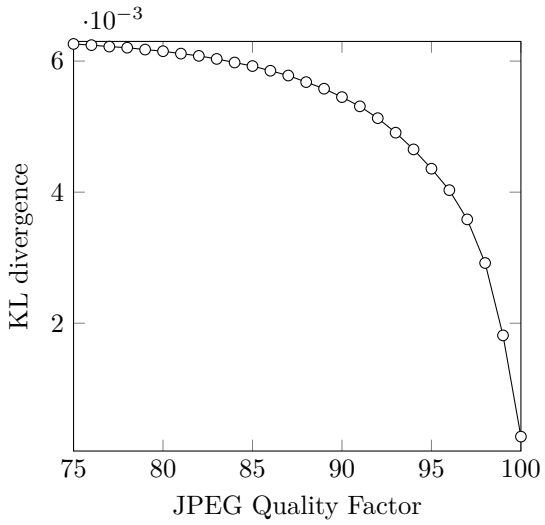

Figure 2.5.1: KL divergence as a function of the QF for J-UNIWARD at 0.4 bpnzac when using non-rounded and non-maximized quantization matrices.

Table 2.2 shows an example of the average change rates $\overline{\beta}_{kl}$ for J-UNIWARD for quality factor 95 and relative payload 0.4 bpnzac. Note that the change rate is the largest for low spatial frequencies and much smaller for high frequencies. This is because the embedding costs of J-UNIWARD are larger for larger quantization steps q_{kl}, which roughly correspond to higher spatial frequencies.

2.5 Experiments

In this section, we report the results of all experiments, which include the accuracy of empirical detectors as a function of the JPEG quality factor for several algorithms and payloads contrasted with the KL divergence computed from the model of JPEG coefficients introduced in Section 2.2. The investigation focuses on the case when the relative payload is fixed in terms of bpnzac because it is far more interesting than for bpp, which we briefly comment upon in Section 2.5.3.

2.5.1 Modern steganography

The initial investigation deals with J-UNIWARD [74]. Two types of empirical detectors were studied: the ensemble classifier [97] with Gabor Filter Residual (GFR) features [119], as a representative of the paradigm of rich models, and the Steganalysis Residual Network (SRNet) [8] as a representative of detectors built using deep learning. Based on the experiments reported in [8], the SRNet currently provides the most accurate detection of modern JPEG steganography over other competing architectures designed for the JPEG domain [130, 140].

Figure 2.3.1 left shows the performance of both detectors for payload 0.4 bpnzac across JPEG qualities 85–100 in terms of the correct classification accuracy $1 - P_{\text{E}}$, where

$$P_{\text{E}} = \min \frac{1}{2}(P_{\text{MD}} + P_{\text{FA}}) \tag{2.5.1}$$

is the often used minimum average detection error under equal priors, and P_{MD} and P_{FA} the missed-detection and false-alarm rates. The right y axis shows the scale of the KL divergence (2.3.16) computed between the cover model (2.2.5) and the stego model of J-UNIWARD (2.3.14). With the exception of GFR for quality 99 and 100, both empirical detectors closely mimic the variations of the

KL divergence across all quality factors, including the small "ripples" at 86, 88, 90, and 93, due to rounding and clipping of the quantization steps (2.2.1) as well as the minimum around quality 95–96. To confirm the origin of the ripples, in Figure 2.5.1 we show the KL divergence for J-UNIWARD at 0.4 bpnzac, when the quantization steps q_{kl} are not rounded to integers and not clipped to 1 (when removing "max" and rounding "[.]" in (2.5.1)) with $\mathbf{q}(100) \triangleq \mathbf{q}(99)/10$ as (2.2.1) would produce a matrix of zeros for quality 100. The KL divergence for J-UNIWARD in this case monotonically and smoothly decreases with increased quality factor Q.

Furthermore, still inspecting Figure 2.3.1, the SRNet provides markedly better detection than GFR. In particular, GFR appears to significantly under-perform w.r.t. the SRNet for quality factors above 98. For the two largest quality factors 99 and 100, the KL divergence predicts that the detection should be much more accurate than what the SRNet exhibits, perhaps indicating a possible space for improvement. Since the SRNet generally offers much better detection than GFR, all remaining experiments, unless otherwise mentioned, are executed with the SRNet as the empirical detector.

In Figure 2.3.1 right, we show the detectability of UED-JC across quality factors for a fixed payload 0.3 bpnzac. The trends of the empirical detector, including the small variations between QF 85 and 91 due to quantization step rounding again closely match the KL divergence computed between the models. As with J-UNIWARD, the KL divergence values seem to suggest that the empirical detector under-performs for qualities near 100.

Before we move to older steganographic paradigms in the next section, we note that for the experiments reported above, the SRNet was initially trained as described in the original publication [8] from randomly initialized filters for quality factor 85 as this is when both J-UNIWARD and UED-JC are the most detectable. Curriculum training via the quality factor was used to train for 86, 87, ..., 100 and was always run for 100k iterations with LR 10^{-3} after which the LR was lowered to 10^{-4} for an additional 50k iterations.

2.5.2 Old steganography

In this section, the relationship between the empirical detection accuracy and the KL divergence between the cover and stego models has been investigated for a generic LSB flipper, OutGuess, model-based steganography (MBS), generic ternary embedding in non-zero DCT coefficients (LSBM), and nsF5. The results are summarized in graphical form in Figure 2.5.2.

In contrast to J-UNIWARD and UED-JC, except for nsF5, all embedding methods exhibit the same qualitative trend – their empirical security decreases with increasing quality factor but this trend eventually reverses for larger quality factors. Since the details of how MBS handles embedding a payload smaller than the maximal payload have not been available to the authors, the KL divergence displayed in the graph showing MBS is for LSBM.

The corresponding KL divergence between the models relatively well matches the empirical results. The nsF5 was the only embedding algorithm for which the KL divergence exhibited a different trend than the empirical detectors (Figure 2.5.2 bottom right). While both the SRNet and the ensemble with GFR exhibit approximately constant detectability, the model predicts an increasing KL divergence. This could mean that either our model fails to capture the impact of embedding correctly for this algorithm or that the empirical detectors increasingly under-perform for larger quality factors. We hypothesize that the latter explanation is more likely for the following reason. The increase of the KL divergence for nsF5 is primarily due to the increased number of zeros in stego images since the embedding operation always decreases the absolute value of DCT coefficients. Detecting this increase or, equivalently, estimating the number of zeros in the cover from the stego image, however, seems to be a difficult task in practice. We intend to investigate this issue as part of our future effort.

For all embedding methods, the SRNet was first trained from scratch for QF 95 because this is the range with the easiest detection. The detectors for the remaining QFs were trained using curriculum training via the quality factor in quality factor steps of one.

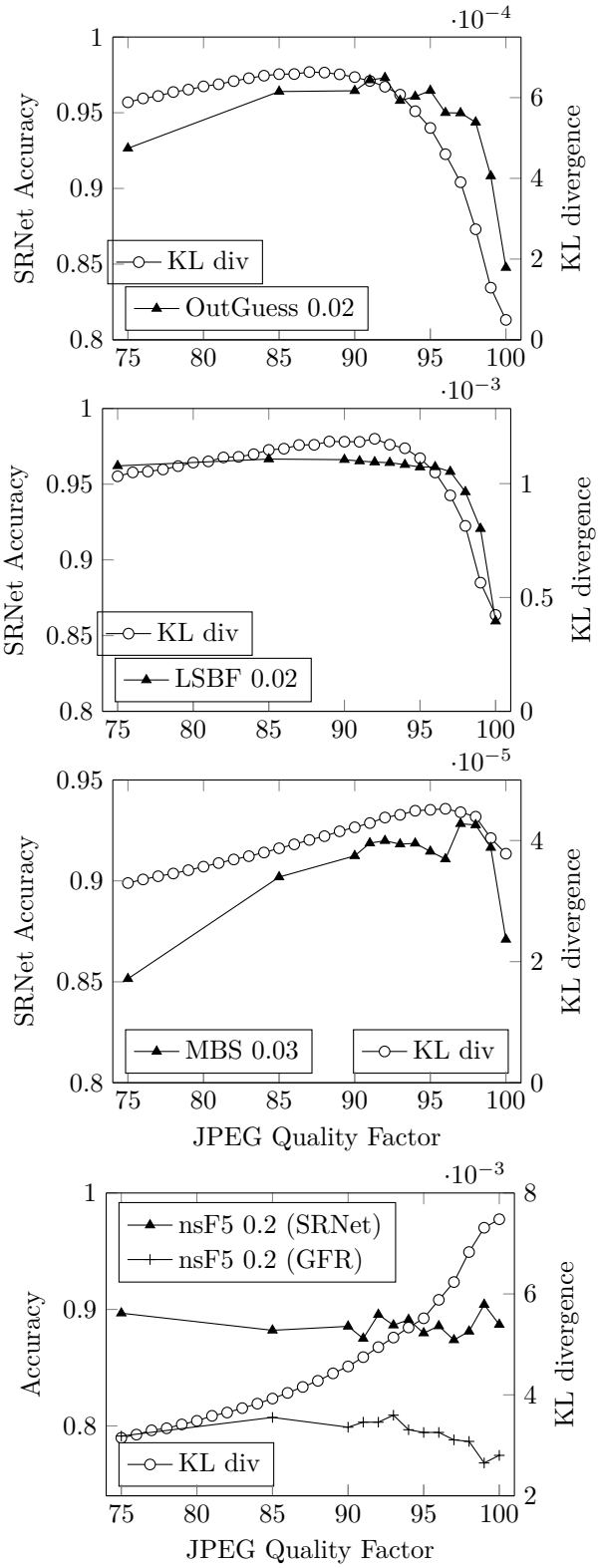

Figure 2.5.2: By rows: Detection accuracy of SRNet for OutGuess at 0.02 bpnzac, LSBF at 0.02 bpnzac, MBS at 0.03 bpnzac, and nsF5 at 0.2 bpnzac (left axis). The right axis is for the KL divergence (2.3.16) between cover and the corresponding stego models.

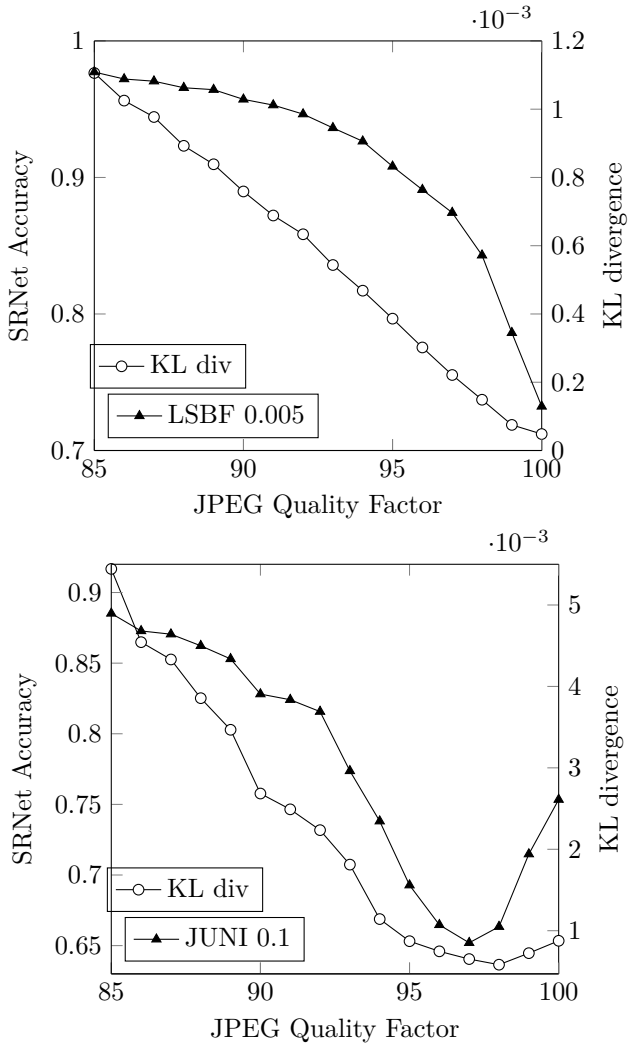

Figure 2.5.3: Accuracy of SRNet and the KL divergence between cover and stego models for LSBF at 0.005 bpp (left) and J-UNIWARD (right) at 0.1 bpp.

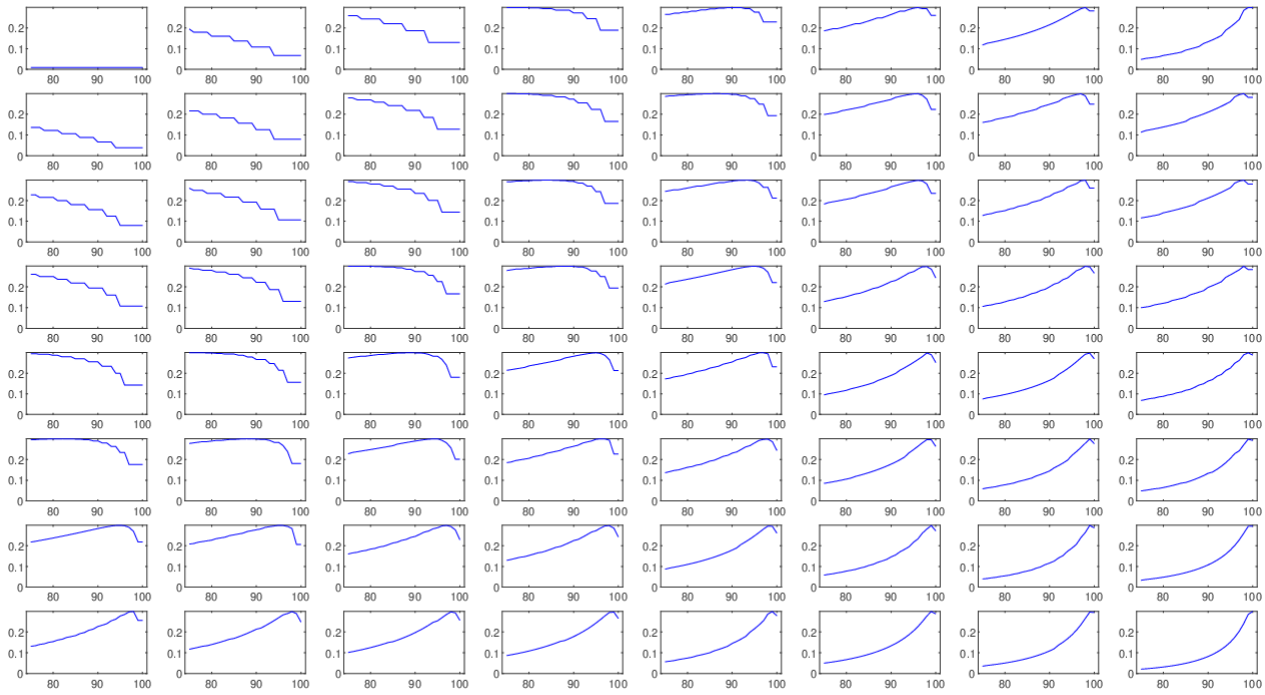

Figure 2.5.4: Fisher information $I_{kl}(Q)$ for generic LSBM as a function of the quality factor $75 \leq Q \leq 100$ for all 64 DCT modes with k and l corresponding to rows and columns, respectively.

2.5.3 Fixed bpp

For completeness, we briefly report the results obtained when the payload is fixed in terms of bits per pixel (bpp) rather than bpnzac. A relative payload fixed in terms of bpp means that the same number of bits is embedded for all quality factors and all steganographic algorithms. Since the number of non-zero DCT coefficients strictly increases with increased quality factor, the "effective size" of the cover for old steganography paradigms increases. Our model predicts a strictly decreasing KL divergence for all old stego methods. As an example, in Figure 2.5.3 left we show the SRNet accuracy and the KL divergence for LSBF at payload 0.1 bpp.

In contrast, for modern steganography, the detectability decreases but starts increasing for qualities close to 100. Figure 2.5.3 right shows the detection accuracy of the SRNet and the KL divergence between cover and stego models for J-UNIWARD when fixing the relative payload at 0.1 bpp. The model correctly predicts the lowest detectability around 97–98 as well as the small "ripples" between 85 and 93.

2.6 Analysis

In this section, we present a more intuitive explanation of the complementary security trends observed for old and modern steganography. This requires inspecting in more detail how the KL divergence of individual DCT modes changes with increasing quality factor. We first study old steganography paradigms and then modern schemes.

2.6.1 Old steganography

We work with the generic LSBM (2.3.13) with global change rate β w.r.t. all non-zero DCT coefficients as this will simplify our arguments. The leading term of the Taylor expansion of the KL divergence (2.3.16) with respect to β is :

$$D_{\mathrm{KL}}(P^{(c)}||P^{(s)}) \doteq \frac{\beta^2}{2}\sum_{k,l=0}^{7} I_{kl}, \tag{2.6.1}$$

where

$$I_{kl} = \sum_{m} \frac{1}{P^{(c)}_{kl}(m)} \left(\frac{\partial P^{(s)}_{kl}(m)}{\partial \beta_{kl}}\Big|_{\beta_{kl}=0} \right)^2. \tag{2.6.2}$$

is the steganographic Fisher information for mode (k,l). Thus, to understand the trends w.r.t. the quality factor Q, we need to inspect I_{kl} as a function of Q. First, we take a look at the range $Q \leq 95$.

Figure 2.5.4 shows $I_{kl}(Q)$ for $75 \leq Q \leq 100$ with the y-axis scale unified across all modes. Note that the Fisher information for low frequency modes decreases, it exhibits a non-monotone trend for medium frequencies, and sharply increases for high frequencies. With increasing Q, the increase in $I_{kl}(Q)$ for high spatial frequencies is larger than the decrease of $I_{kl}(Q)$ for low spatial frequencies, which clarifies the security trend of old embedding methods observed in the previous section. Note that this trend can be reversed by letting the change rates decrease with increasing $k+l$ as is the case for modern steganography.

The seemingly complex behavior of $I_{kl}(Q)$ w.r.t. Q is caused by the fact that old steganography does not embed into zeros. To see why, we point out that the cover p.m.f. $P^{(c)}_{kl}$ (2.2.5) depends only on the ratio $w_{kl}/q_{kl}(Q)$ (see Eq. (2.2.6)), the effective width of the GG model after quantizing the (k,l)th mode with quantization step $q_{kl}(Q)$. Figure 2.6.1 (solid line, right y-axis) shows the Fisher information I as a function of the ratio w/q for $\gamma = 0.4$.[2] Note that I exhibits a maximum at $w/q \approx 0.04$. In contrast, when allowing embedding into zeros, I becomes strictly decreasing w.r.t. Q (the dashed line, left y-axis shows $\log_{10} I$ in Figure 2.6.1). For DCT modes (k,l) for which $w_{kl}/q_{kl}(Q) \leq 0.04$, increasing the quality factor leads to increased Fisher information $I_{kl}(Q)$. This occurs for high frequency modes because the width of their GG fit is smaller (Table 2.1). For modes with $w_{kl}/q_{kl}(Q) \geq 0.04$, $I_{kl}(Q)$ decreases with increased Q. The non-monotone behavior of $I_{kl}(Q)$ for medium frequencies is due to the ratio $w_{kl}/q_{kl}(Q)$ moving past 0.04 as Q increases.

Once $Q > 95$, for low spatial frequencies the quantization steps $q_{kl}(Q)$ start "flattening out" at 1, which means that the ratio $w_{kl}/q_{kl}(Q)$ stops increasing and thus $I_{kl}(Q)$ no longer decreases with Q. The Fisher information of high and medium-frequency modes start decreasing as the ratios $w_{kl}/q_{kl}(Q)$ grow larger than 0.04, eventually reversing the detectability trend for old steganography.

2.6.2 Modern steganography

Explaining the trend reversal for modern steganography is more complicated due to the modulation of change rates across spatial frequencies and their dependence on the quality factor. We can no longer factor out the global change as in (2.6.1) and need to consider $d_{kl}(Q) = \overline{\beta}^2_{kl}(Q) I_{kl}(Q)$ as functions of Q.

Generally speaking, for low-frequency modes, $d_{kl}(Q)$ decrease with increasing Q. For medium and high frequencies, however, d_{kl} starts to sharply increase for $Q > 95$ (see Figure 2.6.2). This rapid increase is responsible for the reversal of the detectability trend observed for modern embedding schemes for high quality factors, which holds for fixed relative payload in both bpnzac and bpp :

[2] From Table 2.1, $\gamma \approx 0.4$ across all AC modes.

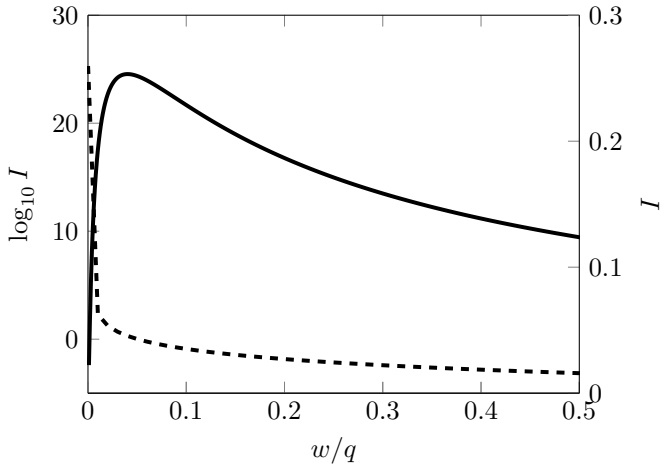

Figure 2.6.1: Solid line and right y-axis: Fisher information I of LSBM as a function of the ratio w/q for $\gamma = 0.4$. Dashed line and left y-axis: Logarithm of Fisher information of LSBM when embedding into zeros.

Attenuation of change rates across modes is not optimal.

This observation gives a clue on a possible improvement of J-UNIWARD, which we briefly delve into in the next section.

2.6.3 Improving J-UNIWARD

In the previous section, we concluded from our model that the increase of detectability (KL divergence) of J-UNIWARD for high quality factors is due to improper modulation of embedding change rates. In particular, the change rates for high spatial frequencies should be attenuated more aggressively than what the embedding distortion of J-UNIWARD dictates. This shows a possible way to improve its security.

Recalling that $\overline{\beta}_{kl}(\alpha, Q)$ is the average change rate applied by J-UNIWARD to mode (k, l) for a given payload α and quality factor Q, we find $\tilde{\beta}_{kl}(\alpha, Q)$, minimizing the leading term of the KL divergence while communicating on average the same entropy:

$$\min_{\tilde{\beta}_{kl}} \sum_{k,l=0}^{7} \tilde{\beta}_{kl}^2(\alpha, Q) I_{kl}(Q) \tag{2.6.3}$$

$$\sum_{k,l=0}^{7} H_3(\tilde{\beta}_{kl}) = \sum_{k,l=0}^{7} H_3(\overline{\beta}_{kl}). \tag{2.6.4}$$

Since the DC term is difficult to model, we avoid optimizing it and instead set $\tilde{\beta}_{00} = \overline{\beta}_{00}$. The change rates $\tilde{\beta}_{kl}$, $k + l > 0$, found in this manner are indeed smaller for high frequencies ($k > 5$ or $l > 5$) and larger for low and medium frequencies. Figure 2.6.3 shows $\overline{\beta}_{kl}$ and $\tilde{\beta}_{kl}$ for $Q = 100$ and relative payload $\alpha = 0.1$ bpp. Note that $\tilde{\beta}_{kl} > \overline{\beta}_{kl}$ for low frequencies and $\tilde{\beta}_{kl} < \overline{\beta}_{kl}$ for high spatial frequencies. Also, while $\tilde{\beta}_{kl}$ decrease with increased frequency, the smallest values of $\overline{\beta}_{kl}$ roughly correspond to the largest entries in $\mathbf{q}(50)$ (see Eq. (8.4.8)).

To incorporate this adjustment into the embedding algorithm, we first convert both $\overline{\beta}_{kl}$ and $\tilde{\beta}_{kl}$ to embedding costs

$$\tilde{\varrho}_{kl} = \ln\left(\frac{1}{\tilde{\beta}_{kl}} - 2\right) \tag{2.6.5}$$

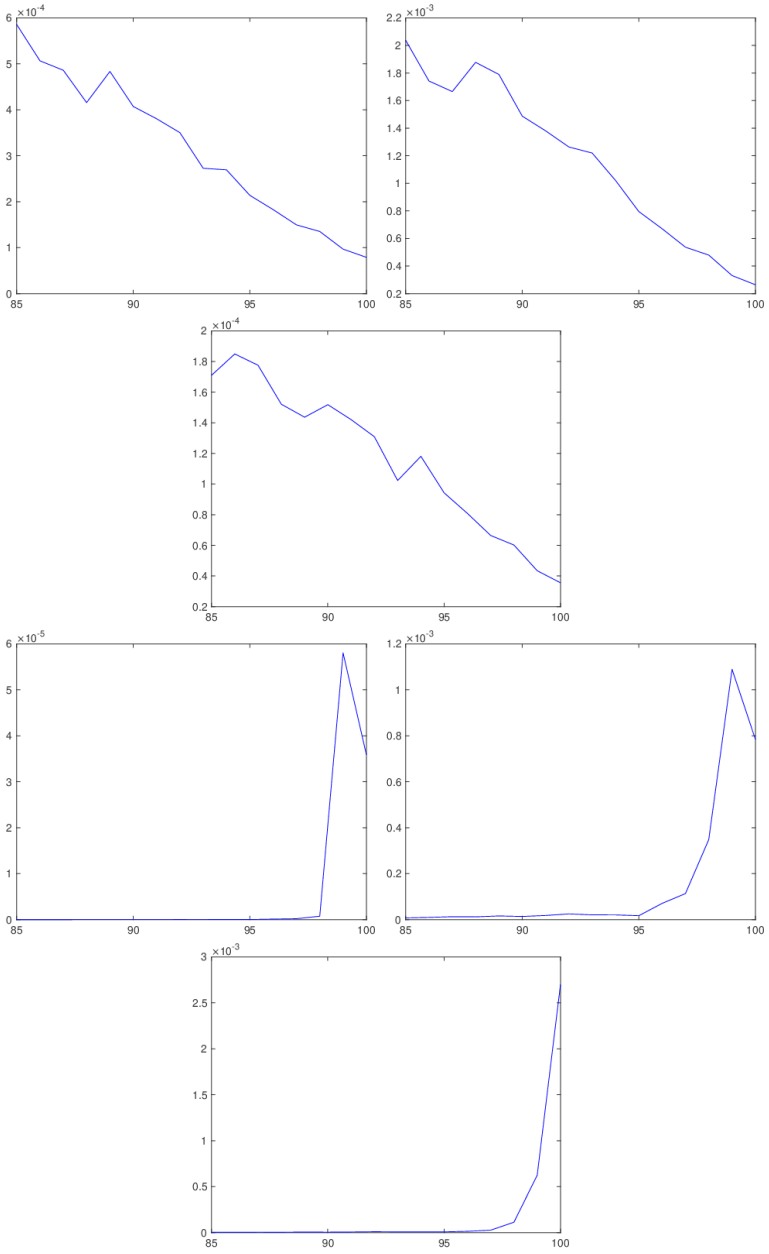

Figure 2.6.2: By rows: the leading term of the KL divergence $d_{kl}(Q)$ in modes $(0,1),(0,2),(1,0),(4,4),(1,7),(7,7)$ as a function of the quality factor Q. J-UNIWARD, 0.4 bpnzac.

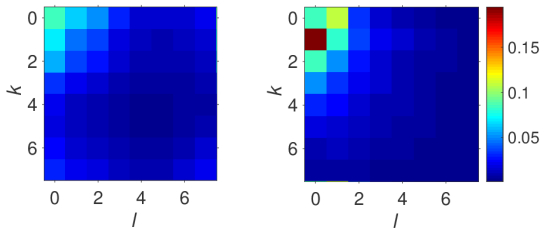

Figure 2.6.3: Left: $\overline{\beta}_{kl}$, right: $\tilde{\beta}_{kl}$ for quality factor $Q = 100$ and relative payload $\alpha = 0.1$ bpp.

$$\overline{\varrho}_{kl} = \ln\left(\frac{1}{\overline{\beta}_{kl}} - 2\right). \tag{2.6.6}$$

Given the matrix of J-UNIWARD's embedding costs in (a, b)th 8×8 block of image $\mathbf{x}$ as $\rho_{kl}^{(a,b)}(\mathbf{x})$, we modulate them

$$\rho_{kl}^{(a,b)}(\mathbf{x}) \to \rho_{kl}^{(a,b)}(\mathbf{x})\frac{\tilde{\varrho}_{kl}}{\overline{\varrho}_{kl}}. \tag{2.6.7}$$

These modulated costs would then be used in an embedding simulator or STCs for practical embedding in image $\mathbf{x}$. Note that the modulation (2.6.7) depends on payload α as well as the quality factor Q.

This heuristic adjustment of the embedding change rates did indeed improve J-UNIWARD's security. For quality factor 100, the accuracy of SRNet decreased by 2.14%. The network detector was trained by seeding with detector trained on J-UNIWARD for $\alpha = 0.1$ bpp and the corresponding quality factor. The LR was 10^{-3} for the first 100k iterations, lowered to 10^{-4} for an additional 50k iterations.

To further validate this approach, we carried out the same experiment for quality factors 100 for J-UNIWARD at $\alpha = 0.4$ bpnzac. In this setting, the security was improved by 1.12% in terms of SRNet accuracy.

Due to limited space and time, we postpone a more detailed investigation to our future work. In particular, more detailed study needs to be executed regarding the change rate adjustment across payloads and quality factors as well as for other embedding schemes. The limited experiment in this section should thus be thought of more as a promising direction and additional evidence for the predictive power of our theoretical approach.

2.7 Conclusions

This paper investigates how the detectability of JPEG steganography changes with the quality factor when fixing the relative payload. While older embedding paradigms become progressively more detectable up until quality 90–95 after which their detectability decreases, modern steganography exhibits complementary trends. This behavior is explained by modeling a JPEG file as 64 independent channels with a quantized generalized Gaussian distribution. The KL divergence between cover and stego distributions closely matches the detectability obtained with empirical detectors. The only tested algorithm for which our theoretical analysis failed to match the results of empirical detectors is nsF5. We hypothesize that this is due to the inability of empirical detectors to assess the number of zeros in a JPEG file, indicating a possible improvement of detection of nsF5 for larger quality factors.

By analyzing the Fisher information as a function of the width of the GG model, we offer a more intuitive explanation of the observed trends. For old embedding paradigms, the contribution of

high-frequency modes to detectability increases faster with increased quality than the decrease in detectability in low-frequency modes. This trend can be reversed by decreasing the change rates with increased spatial frequency. For modern steganography, the loss of security of J-UNIWARD for high quality factors has been linked to slightly improper modulation of change rates across spatial frequencies. A heuristic adjustment of the change rates based on the insight obtained from the model indeed lead to an improved security of J-UNIWARD for quality factor 100.

A by-product of our analysis is a better understanding of why older embedding paradigms are much less secure than modern schemes: the comparatively large change rates for high-frequency modes in older schemes substantially increase the KL divergence but contribute little to the total payload because they contain fewer non-zero coefficients. Modern steganography addresses this problem by decreasing the change rate with increasing spatial frequency.

Numerous imaging devices and image editing software use non-standard quantization matrices, which were not investigated in this work. However, the authors are fairly confident that the findings of this paper qualitatively generalize to custom quantization matrices with respect to a generalized concept of JPEG quality defined by a suitably chosen distance (metric) between quantization matrices.

Despite the fact that our model cannot properly capture content adaptivity of modern steganography, its predictive power allowed us to explain the security trends w.r.t. JPEG quality factor and improve the security of J-UNIWARD for the largest quality factor. The use of the model for steganography is a topic that deserves a more extensive study and is thus left for future research.

Chapter 3

Steganography and its Detection in JPEG Images Obtained with the "Trunc" Quantizer

Many portable imaging devices use the operation of "trunc" (rounding towards zero) instead of rounding as the final quantizer for computing DCT coefficients during JPEG compression. We show that this has rather profound consequences for steganography and its detection. In particular, side-informed steganography needs to be redesigned due to the different nature of the rounding error. The steganographic algorithm J-UNIWARD becomes vulnerable to steganalysis with the JPEG rich model and needs to be adjusted for this source. Steganalysis detectors need to be retrained since a steganalyst unaware of the existence of the trunc quantizer will experience 100% false alarm.

3.1 Introduction

Steganography in JPEG images is usually executed by partially decompressing the JPEG file and modifying the quantized DCT coefficients by at most ± 1. To the best of the authors' knowledge, the entire bulk of previous art on JPEG steganography assumes that the last step of JPEG compression involves rounding the DCT coefficients to the nearest integer [108]. Such JPEG images will be referred to as coming from the *round source.* As recently pointed out in [3], however, many modern portable imaging devices, such as iPhone 5c, Canon EOS 10D, Samsung Galaxy Tab 3 8.0, replace the rounding with "rounding towards zero" due to its easier (more efficient) hardware implementation. We will refer to JPEG images processed this way as coming from the *trunc source.*

This paper studies both steganography and steganalysis in trunc JPEGs. In the next section, we introduce notation, datasets, and the setup of experiments. In Section 3.3, we show that a steganalyst unaware of the existence of the trunc quantizer will experience 100% false alarm rate independently of the steganography and the detector. We also show that steganography in trunc JPEGs is more secure. In Sections 3.4 and 3.5, J-UNIWARD and SI-UNIWARD stego algorithms are redesigned to reflect the specifics of the new source. The paper is concluded in Section 7.5.

3.2 Preliminaries

For simplicity, we only work with 8-bit grayscale images. Pixel values and unquantized DCT coefficients in an 8×8 block will be denoted x_{ij} and d_{kl}, $0 \leq i, j, k, l \leq 7$. The classical rounding

operation will be denoted $\mathcal{Q}_{round}(x) = \lfloor x \rceil$ while the trunc quantizer is $\mathcal{Q}_{trunc}(x) = \lfloor x \rfloor$ for $x \geq 0$ and $\mathcal{Q}_{trunc}(x) = \lceil x \rceil$ for $x < 0$, where $\lfloor x \rfloor$ and $\lceil x \rceil$ represent flooring and ceiling. Quantized DCT coefficients are $\mathcal{Q}_{(\cdot)}(d_{kl}/q_{kl})$, where q_{kl} are the luminance quantization steps. The rounding error during compression is defined as $e_{kl} = c_{kl} - \mathcal{Q}_{(\cdot)}(c_{kl})$, where we denoted $c_{kl} = d_{kl}/q_{kl}$.

All experiments are carried out on the union of BOSSbase 1.01 and BOWS2 datasets, each with 10,000 grayscale images, resized from their original size 512×512 to 256×256 using `imresize` with default setting in Matlab. Cover JPEG images coming from the round source were obtained in Matlab using the command `imwrite`. Cover JPEG images from the trunc source were obtained in Matlab by applying Matlab's `dct2` on blocks of pixels, dividing the coefficients by the quantization matrix, applying the trunc quantizer $\mathcal{Q}_{trunc}(x)$, and saving them to a JPEG file using Phil Sallee's `jpeg_write`. Decompression to the spatial domain for experiments with empirical detectors was obtained by multiplying the DCT coefficients by quantization steps and applying a block inverse DCT without rounding or clipping in Python by applying 'fftpack.idct' with the parameter norm = 'ortho', from Python's SciPy library, horizontally and vertically.

For training empirical detectors, we randomly selected 4,000 images from BOSSbase and the entire BOWS2 dataset with 1,000 BOSSbase images set aside for validation. The remaining 5,000 BOSSbase images were used for testing. In summary, $2 \times 14,000$ cover and stego images were used for training, $2 \times 1,000$ for validation, and $2 \times 5,000$ for testing. This dataset and the split into training and testing has also been used in [133, 135, 8].

For steganalysis, we selected the SRNet [8], the cartesian-calibrated JPEG Rich Model (ccJRM) [96], and Gabor Filter Residual features (GFR) [119] with the FLD ensemble [97]. The ensemble was trained on the union of the training and validation sets. For training SRNet from scratch, we set the initial learning rate (LR) to 10^{-3} for 400k iterations and continued for 100k more iterations with LR 10^{-4} and batch size 32. When seeding, we use LR 10^{-3} for 100k iterations and a lower the LR to 10^{-4} for additional 50k iterations.

3.3 Comparing the sources

For experiments in this section, we selected the steganographic algorithm nsF5 [57] with relative payload 0.2 bpnzac, J-UNIWARD [74] with payload 0.4 bpnzac, and UED [66, 67] with payload 0.3 bpnzac.

3.3.1 Quantizer mismatch

First, we study what happens when the detector is unaware of the existence of the trunc source and uses a detector trained on the round source for steganalysis of trunc JPEGs. Experiments were executed with three different detectors for quality factors 85 and 100 and various steganographic algorithms and payloads. To be more specific, we trained a classifier for a given stego algorithm (and fixed payload) on cover and stego images from the round source. This detector was then tested on cover and stego JPEGs embedded with the same stego algorithm and payload but starting with trunc JPEG covers instead. The end result was always the same – both cover and stego images from the trunc source were detected as stego irrespectively of the embedding algorithm, payload and detector, with the false alarm rate ranging between 99.1% and 100%.

Fortunately, it is easy to reliably identify the type of the DCT quantizer and build separate detectors for each source. Table 3.1 shows the accuracy of a classifier trained on two classes: cover JPEG images coming from the trunc and round source for quality 85 and 100. The training was stopped after 70k iterations, since the validation accuracy already saturated at 100%. Note that this detector correctly reveals the DCT quantizer even when presented with stego images embedded with various payloads and different stego algorithms. Having this classifier, from now on we will assume that the steganalyst knows whether an image under investigation comes from the round or the trunc source.

Algorithm	Payload	QF85	QF100
Covers	0	0.9999	0.9989
nsF5	0.2	0.9998	0.9987
JUNI	0.4	0.9999	0.9987
UED	0.3	0.9997	0.9987

Table 3.1: Accuracy of detecting the DCT quantizer. The detector is the SRNet trained between cover classes from the round and trunc sources and tested on 5,000 pairs of images from each of the four sources.

3.3.2 Effect of truncation on security

Since the histogram bin for zero coefficients in the trunc source is twice as wide as all other bins, cover images in trunc source have more zeros than covers in the round source. For a fixed image, its "effective" size, the number of non-zero DCT coefficients [89], is smaller in the trunc source than in the round source. For a fair comparison of the security of a given stego algorithm in both sources, we thus adjust the size of the embedded payload according to the square root law [89, 42, 87, 86]. The relative payload in the trunc source, α_{trunc}, was scaled as

$$\alpha_{trunc} = \alpha_{round} \cdot \sqrt{\frac{N_{round}}{N_{trunc}}} \cdot \frac{\log(N_{trunc})}{\log(N_{round})}, \tag{3.3.1}$$

where N_{trunc} and N_{round} stand for the number of non-zero AC DCT coefficients from a given image in trunc and round sources, respectively. The accuracy[1] shown in Figure 3.3.1 were obtained with three different detectors: SRNet and the ensemble classifier with JRM and GFR features on the same embedding algorithms and payloads as above. SRNets on quality 75 were trained from scratch, while 95 was trained via curriculum training from 75. For nsF5, the network was first trained on quality 95 from scratch and then retrained for the smaller quality because the higher quality is more detectable [12]. Note that even with the scaled payload, the detection accuracy is larger in the round source across all algorithms and detectors, indicating that it is harder to detect steganography in the trunc source. A *surprising exception* is J-UNIWARD, which is best detected in trunc source with JRM. As shown in the next section, this is because J-UNIWARD embeds "too much" into zero DCT coefficients, which are much more populated in the trunc source, and consequently introduces artifacts detectable by JRM.

3.4 J-UNIWARD for trunc source

As mentioned in the previous section, J-UNIWARD in the trunc source is best detected with the JRM because it embeds too much into zero coefficients. Figure 3.4.1 top left shows that stego images have significantly fewer zero coefficients than cover images. This lead us to the following adjustment of the embedding algorithm.

For a fixed DCT mode (k, l), let β_i be the average J-UNIWARD change rate on such coefficients that are equal to i in the cover image. If there are no coefficients equal to i, we set $\beta_i = 0$. Let ρ_i be the corresponding "average cost"

$$\rho_i = 1/\lambda \log(1/\beta_i - 2), \tag{3.4.1}$$

where $\lambda > 0$, is a Lagrange multiplier. We wish to adjust $\rho_0 \to \tilde{\rho_0} = \eta\rho_0$, $\eta > 0$, so that the new change rate of zeros

[1] Accuracy for the ensemble with rich models is computed as $1 - P_{\mathrm{E}}$, where P_{E} is the minimum average total probability of error.

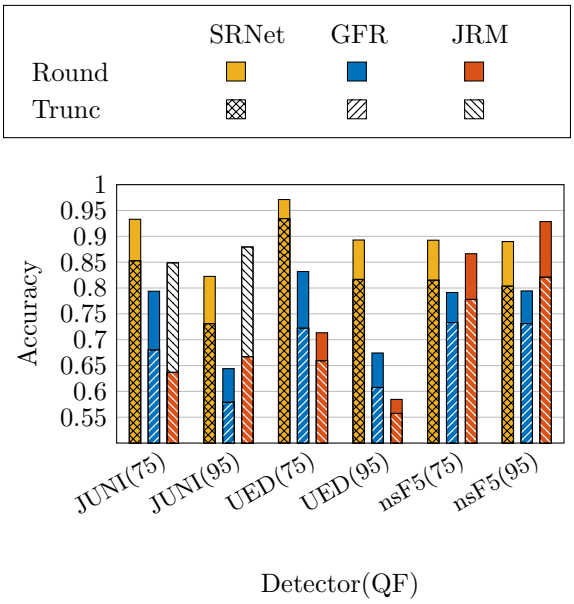

Figure 3.3.1: Detection accuracy in the trunc source and the round source when adjusting for the square root law for J-UNIWARD, UED, and nsF5 with relative payloads 0.4, 0.3, and 0.2 bpnzac.

$$\tilde{\beta}_0 = \frac{e^{-\lambda\rho_0\eta}}{1+2e^{-\lambda\rho_0\eta}} \tag{3.4.2}$$

preserves on average the number of zero coefficients:

$$(1-2\tilde{\beta}_0)h[0] + \beta_{-1}h[-1] + \beta_1 h[1] = h[0], \tag{3.4.3}$$

where $h[i]$ is number of cover coefficients equal to i. Assuming $\beta_{-1} = \beta_1$ and using $\log(1+z) \approx z$ for small $z > 0$, (3.4.2) and (3.4.3) give

$$\eta = \frac{\rho_1}{\rho_0} + \frac{1}{\lambda\rho_0}\log\left(\frac{2h[0]}{h[1]+h[-1]}\right). \tag{3.4.4}$$

Computing the average change rate on coefficients equal to 1 or -1, $\beta_{|1|} = (\beta_{-1}+\beta_1)/2$, from (3.4.1) and (3.4.4)

$$\eta = \frac{\log(1/\beta_{|1|}-2)}{\log(1/\beta_0-2)} + \frac{\log\left(\frac{2h[0]}{h[1]+h[-1]}\right)}{\log(1/\beta_0-2)}. \tag{3.4.5}$$

Technically, the change rates in (3.4.3) have a different Lagrange multiplier because, first, J-UNIWARD simulator is used to compute the average change rates β_i, the costs of zero coefficients are then updated, and a new Lagrange multiplier needs to be found to satisfy the payload constraint. As indicated by our experiments, however, the new Lagrange multiplier produced by such modulation of costs on zero coefficients is almost identical to the original one, justifying thus our simplified approach. Note that, if there are no coefficients equal to 1 or -1, the update rule (3.4.4) naturally sets the costs to wet costs. We call this scheme J-UNIWARD with histogram correction (hcJ-UNIWARD). Figure 3.4.1 top right shows that the embedding indeed roughly preserves the number of zero coefficients.

To show that hcJ-UNIWARD is more secure than J-UNIWARD across quality factors, we trained the SRNet, the ensemble classifier with JRM features, as well as the concatenation of JRM and the

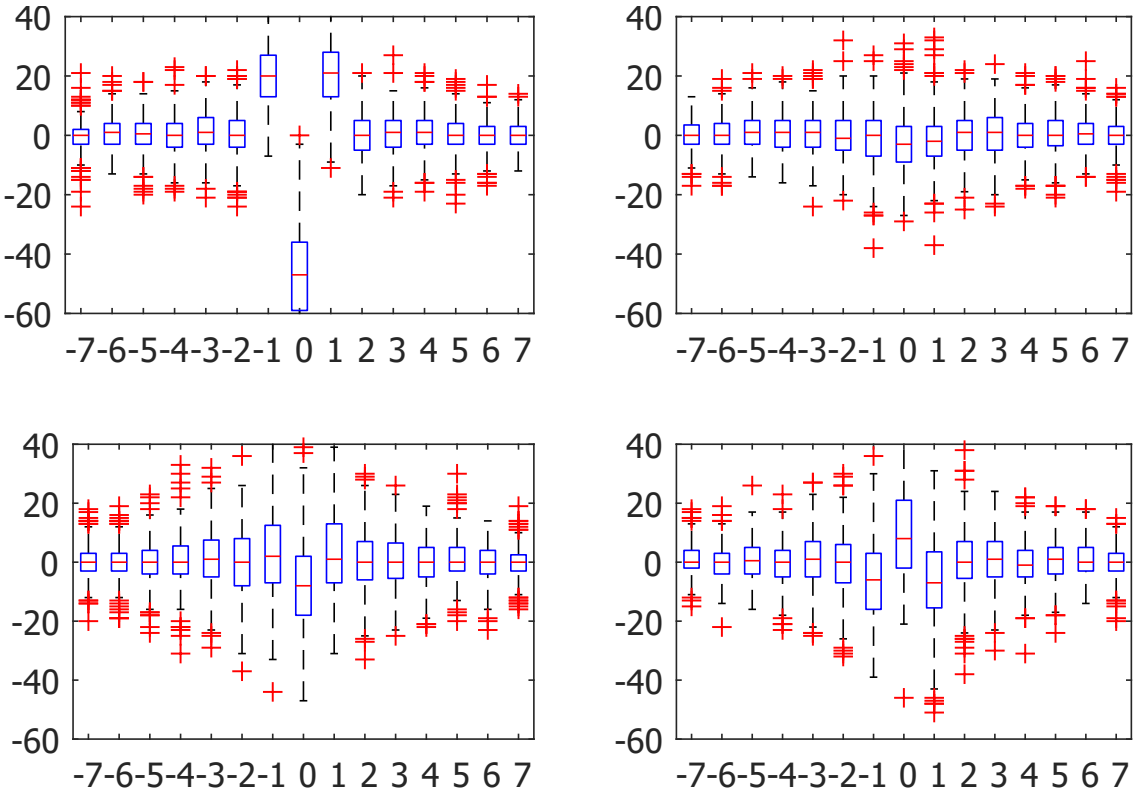

Figure 3.4.1: Boxplots showing the differences between stego (0.4 bpnzac) and cover histograms of DCTs across 300 randomly selected images. From left to right by rows: J-UNIWARD, hcJ-UNIWARD, SI-UNIWARD, hcSI-UNIWARD.

features extracted by the SRNet (the 512-dimensional input to the IP layer) with the low-complexity linear classifier [26]. The improvement in security ranges from $7-15\%$ in terms of accuracy of the best detector among the three detectors mentioned above (see Figure 3.5.1).

3.5 Side information

In side-informed JPEG steganography, the rounding errors during the quantization of DCT coefficients are used to modulate the embedding costs by $1-2|e_{kl}|$. In trunc source, however, the rounding errors have a different range, and the modulation has to be adjusted. Note that a modulation by $1-2|e_{kl}|$ would lead to negative costs. Moreover, it does not correspond to what one would intuitively expect because zero cost should be associated with $e_{kl} \approx 0$ and $|e_{kl}| \approx 1$. In this section, we focus on the ternary version of SI-UNIWARD [74, 31].

We propose to modulate by the minimum perturbation of the precover that makes it quantize to the desired stego value. Denoting $\rho_{kl}(-1)$, $\rho_{kl}(+1)$ J-UNIWARD's costs of changing the kl-th DCT coefficient by -1 and $+1$, respectively, the side-information modulated costs ρ'_{kl} for cover DCT coefficients c_{kl} that quantize to a non-zero integer ($|c_{kl}| \geq 1$)

$$\rho'_{kl}(\text{sign}(e_{kl})) = (1-|e_{kl}|)\rho_{kl} \tag{3.5.1}$$
$$\rho'_{kl}(-\text{sign}(e_{kl})) = |e_{kl}|\rho_{kl} \tag{3.5.2}$$

and for those that quantize to 0 ($|c_{kl}| < 1$)

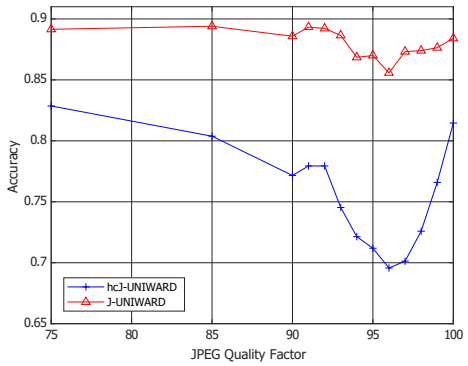

Figure 3.5.1: Accuracy of the best detector in trunc source for hcJ-UNIWARD and J-UNIWARD at 0.4 bpnzac.

bpnzac	1	0.8	0.6	0.4
QF75	0.8164	0.7436	0.6485	0.5653
QF95	0.7984	0.6972	0.6050	0.5420

Table 3.2: Detection accuracy of SRNet for various payloads of ternary SI-UNIWARD in the trunc source.

$$\rho'_{kl}(+1) = (1 - e_{kl})\rho_{kl} \tag{3.5.3}$$
$$\rho'_{kl}(-1) = (1 + e_{kl})\rho_{kl}. \tag{3.5.4}$$

This makes good intuitive sense because when either $e_{kl} \approx 0$ or $|e_{kl}| \approx 1$ the non-quantized coefficient c_{kl} is the most sensitive to noise and should be given a small cost (modulation close to zero). The separate treatment for coefficients that quantize to zero is necessary because the quantization bin for zero is twice as large in the trunc source. Indeed, when $0 < c_{kl} < 1$, $e_{kl} = c_{kl}$, and it takes a perturbation of $1 - e_{kl}$ to quantize to 1 and $1 + e_{kl}$ to quantize to -1.

SI-UNIWARD was implemented and tested in the trunc source for quality factors 75 and 95 at $1, 0.8, 0.6$, and 0.4 bpnzac. Starting with the largest payload, curriculum learning was used to train on the next smaller payload. Detection accuracy of the SRNet is shown in Table 3.2. For the smallest tested payload, the algorithm is practically undetectable, which validates the proposed modulation of costs. Only SRNet's accuracy is shown because the detection power of the ensemble classifier with JRM features was substantially worse.

We also implemented SI-UNIWARD with histogram correction (hcSI-UNIWARD) in the same way we implemented hcJ-UNIWARD, only this time, the modulation factor (3.4.5) was computed with SI-UNIWARD's average change rates. This, however, decreased the security by about 4%. We hypothesize that the modulation of costs of zeros in SI-UNIWARD (3.5.3)–(3.5.4) already addresses the problem with embedding into zeros too much because the total cost of changing a zero is $\rho'_{kl}(+1) + \rho'_{kl}(-1) = 2\rho_{kl}$, while for coefficients that quantize to a non-zero value, this sum is ρ_{kl}. This is supported by the box plots in Figure 3.4.1 bottom.

3.6 Conclusions

JPEG compressors that use rounding towards zero (trunc) instead of rounding are common in portable electronic devices. This quantizer has profound implications for steganography. Steganalyst

unaware of the existence of such a source will experience 100% false alarms. The "trunc JPEGs" are more friendly to steganography than "round JPEGs" even when adjusting the payload according to the square root law. Moreover, and surprisingly, J-UNIWARD's embedding is faulty in trunc JPEGs as it embeds too much into zeros. We describe an effective fix for this problem. Finally, we also propose a novel modulation of costs for side-informed steganography in trunc JPEGs.

Chapter 4

Minimum Perturbation Cost Modulation for Side-Informed Steganography

A new rule for modulating costs in side-informed steganography is proposed. The modulation factors of costs are determined by the minimum perturbation of the precover to quantize to the desired stego value. This new rule is contrasted with the established way of weighting costs by the difference between the rounding errors to the cover and stego values. Experiments are used to demonstrate that the new rule improves security in ternary side-informed UNIWARD in JPEG domain. The new rule arises naturally as the correct cost modulation for JPEG side-informed steganography with the "trunc" quantizer used in many portable digital imaging devices.

4.1 Introduction

Side-informed steganography is a form of covert communication in which a secret message is embedded in a cover object during processing (or conversion) of a precover [82] to cover. For example, the sender can make use of the fact that she has the uncompressed image before applying JPEG compression. In this case, the rounding errors e_{ij} during quantization of DCT (Discrete Cosine Transform) coefficients form the side-information. The actual embedding of the secret message occurs jointly during processing the precover. Intuitively, DCT coefficients with rounding errors $|e_{ij}| \approx 1/2$ are the most "unstable" in the sense that a small amount of noise could cause them to be rounded to a different value during compression. In side-informed steganography, such coefficients are given a smaller embedding cost to minimize the overall statistical impact of embedding changes.

Side-information can have many forms and can be applied whenever a high quality precover is available to the sender who applies to it some information-reducing processing to obtain the cover as long as the last step of the processing is quantization. Examples include converting a true-color image to a palette format [45], JPEG recompression [50], and the by far most popular case of JPEG compression [91, 113, 127, 76, 74, 67, 31].

Originally, side-informed schemes were inherently *binary* in the sense that the only embedding changes allowed were those in which the cover element (before rounding) was "rounded to the second closest value." The authors of [31] showed the benefit of ternary embedding by allowing embedding changes by ± 1 with appropriately modulated costs. The authors noted that the benefit of ternary embedding over binary is larger for fine quantization, e. g., in the spatial domain, and comparatively much smaller for harsh quantization (in JPEG domain). As this paper indicates, this is likely due to

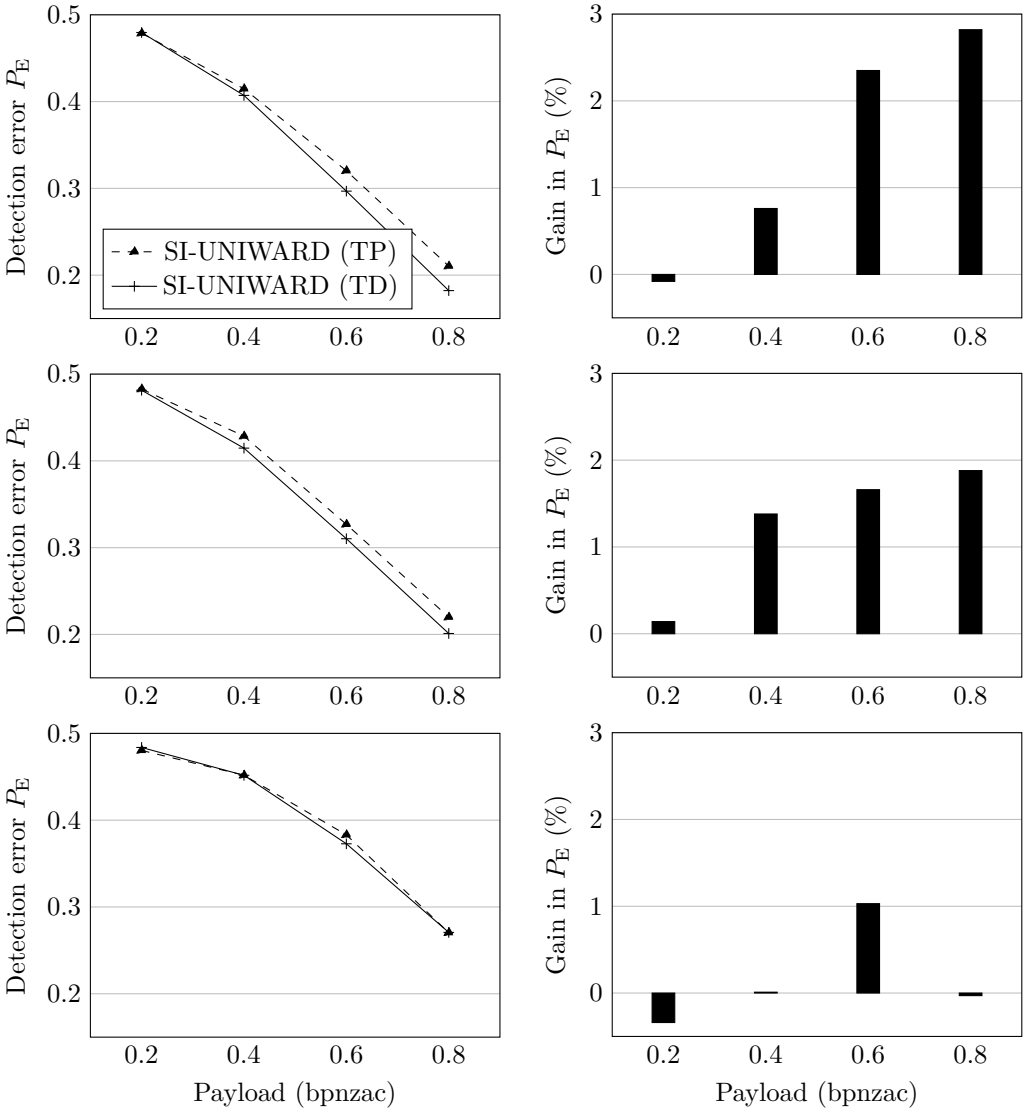

Figure 4.1.1: By rows: Detection error P_E of SCA-GFR / GFR for SI-UNIWARD with (old) cost modulation by difference (TD) (solid) and (new) modulation by minimum perturbation (TP) (dashed) at quality factors 75, 85, and 95 (left). The right column shows the increase of P_E when going from (TD) to (TP) modulations.

not penalizing the cost of the "furthest" (third) stego value enough. To this end, we propose a new heuristic rule for modulating costs based on the minimum perturbation that needs to be applied to the precover to round to the desired stego value. The benefit of this rule is especially apparent when the quantization is harsh. It also universally applies when the quantizer is simple rounding as well as when the quantizer is truncation towards zero as is the case for some JPEG compressors.

In the next section, we review previous art in binary and ternary side-informed steganography. In the third section, we introduce the new rule for cost modulation, and the following section contains the results of all experiments and their interpretation. The paper is concluded in the last section.

4.2 Modulating costs (prior art)

For steganography designed to minimize costs (embedding distortion), a popular heuristic to incorporate a precover value $x_{ij} \in \mathbb{R}$ during embedding is to modulate the costs based on the quantization error, which is in case of rounding, $e_{ij} = x_{ij} - [x_{ij}]$, $-1/2 \leq e_{ij} \leq 1/2$ [91, 127, 76, 67, 74, 31, 113], where $[\cdot]$ denotes the operation of rounding to the nearest integer.

4.2.1 Binary side-informed embedding

A binary embedding scheme modulates the cost of changing $c_{ij} = [x_{ij}]$ to $[x_{ij}]+\text{sign}(e_{ij})$ by $1-2|e_{ij}|$, while prohibiting the change to $[x_{ij}] - \text{sign}(e_{ij})$:

$$\rho_{ij}^{(\text{B})}(\text{sign}(e_{ij})) = (1 - 2|e_{ij}|)\rho_{ij} \tag{4.2.1}$$

$$\rho_{ij}^{(\text{B})}(-\text{sign}(e_{ij})) = \Omega, \tag{4.2.2}$$

where $\rho_{ij}^{(\text{B})}(u)$ is the cost of modifying the cover value by $u \in \{-1, 1\}$, ρ_{ij} are costs of some additive embedding scheme, and Ω is a large constant ("wet" cost [52]). The superscript B indicates that the costs are for binary embedding. This modulation is usually justified heuristically because when $|e_{ij}| \approx 1/2$, a small perturbation of x_{ij} could cause c_{ij} to be rounded to the other side. Such coefficients are thus assigned a proportionally smaller cost because $1 - 2|e_{ij}| \approx 0$. On the other hand, the costs are unchanged when $e_{ij} \approx 0$.

The factor $1 - 2|e_{ij}|$ for cost modulation has been studied in [33], where the authors showed that, based on a discrete Gaussian precover model, the steganographic Fisher information should be modulated by the square of the same factor. This provides some justification to the heuristics in this paper and also in previous art.

4.2.2 Ternary side-informed embedding

A ternary version of this embedding strategy [31] allows modifications both ways with costs :

$$\rho_{ij}^{(\text{TD})}(\text{sign}(e_{ij})) = (1 - 2|e_{ij}|)\rho_{ij} \tag{4.2.3}$$

$$\rho_{ij}^{(\text{TD})}(-\text{sign}(e_{ij})) = \rho_{ij}. \tag{4.2.4}$$

The modulation factors $1 - 2|e_{ij}|$ and 1 are the differences between the rounding errors to a stego element $y_{ij} \in \{[x_{ij}] - 1, [x_{ij}] + 1\}$ and to the cover element :

$$\eta_{ij} = |y_{ij} - x_{ij}| - |x_{ij} - [x_{ij}]|. \tag{4.2.5}$$

Indeed, when $y_{ij} = [x_{ij}]+\text{sign}(e_{ij})$, $\eta_{ij} = 1-2|e_{ij}|$. When $y_{ij} = [x_{ij}]-\text{sign}(e_{ij})$, $\eta_{ij} = 1+|e_{ij}|-|e_{ij}| = 1$, in agreement with (4.2.3)–(4.2.4). The superscript TD stands for **T**ernary cost modulation by **D**ifference. Note that the cost either stays the same or *decreases*, while the sum of both costs is

$$\rho_{ij}^{(\text{TD})}(-1) + \rho_{ij}^{(\text{TD})}(+1) = 2\rho_{ij} - 2|e_{ij}|\rho_{ij}, \tag{4.2.6}$$

and is thus dependent on the rounding error e_{ij}. In the next section, we replace the rule with an alternative rule that assigns larger costs to changes by $-\text{sign}(e_{ij})$, while it assigns the same cost to changes by $\text{sign}(e_{ij})$ as in (4.2.3).

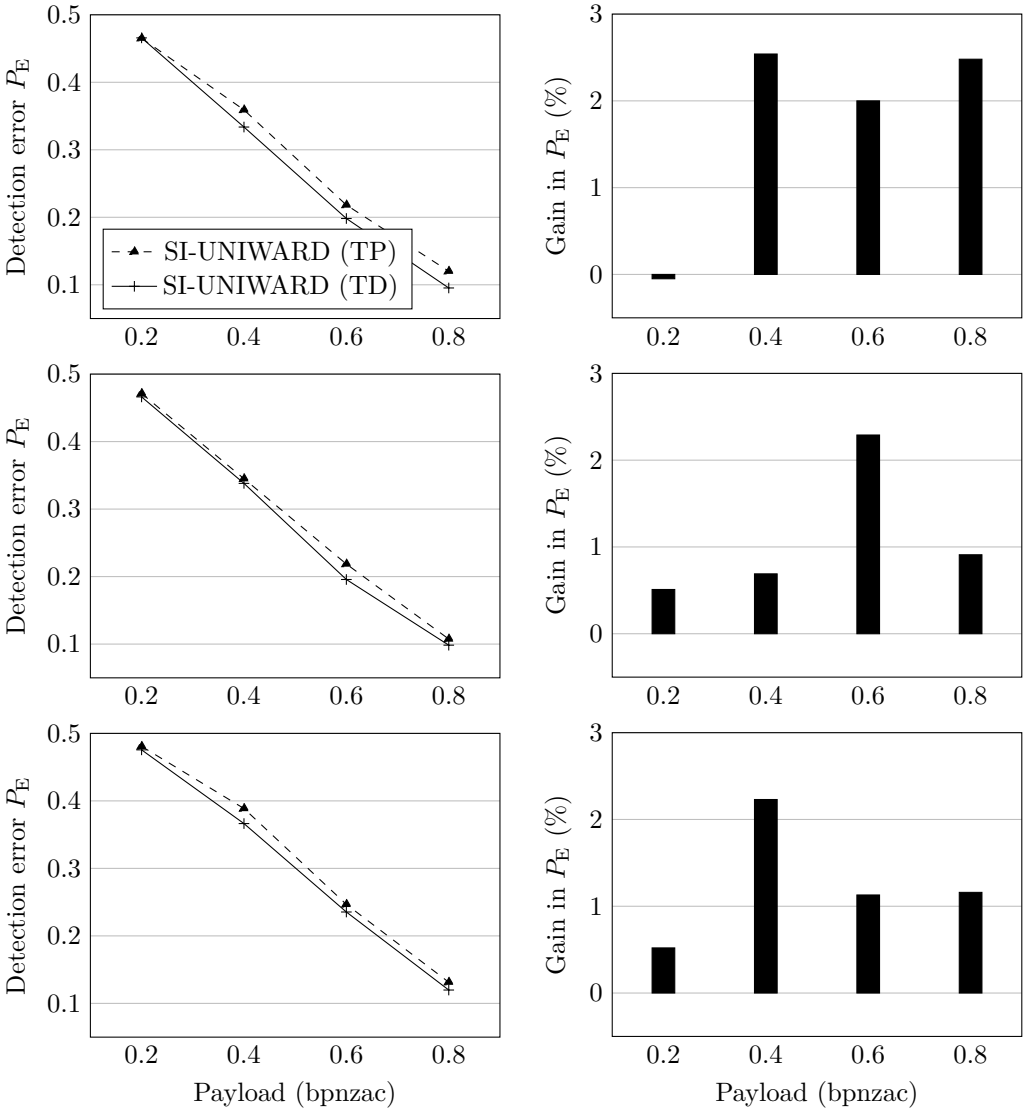

Figure 4.2.1: By rows: Detection error P_E of SCA-SRNet / SRNet for SI-UNIWARD with (old) cost modulation by difference (TD) (solid) and (new) modulation by minimum perturbation (TP) (dashed) at quality factors 75, 85, and 95 (left). The right column shows the increase of P_E when going from (TD) to (TP) modulations.

4.3 Cost modulation by minimum perturbation

The proposed rule can be simply worded in English by stating that the modulation factor is the minimum amount of perturbation applied to the precover to quantize to the desired value. This minimum perturbation is $1/2 - |e_{ij}|$ for change $[x_{ij}] \to [x_{ij}] + \text{sign}(e_{ij})$ and $1/2 + |e_{ij}|$ for $[x_{ij}] \to [x_{ij}] - \text{sign}(e_{ij})$:

$$\rho_{ij}^{(\text{TP})}(\text{sign}(e_{ij})) = (1/2 - |e_{ij}|)\rho_{ij} \tag{4.3.1}$$

$$\rho_{ij}^{(\text{TP})}(-\text{sign}(e_{ij})) = (1/2 + |e_{ij}|)\rho_{ij}. \tag{4.3.2}$$

The superscript TP stands for **T**ernary cost modulation by minimum **P**erturbation. Since multiplying all costs by the same scalar does not change the properties of the embedding scheme, notice

that the modulation factors can equivalently be $1 - 2|e_{ij}|$ and $1 + 2|e_{ij}|$, respectively. In contrast to the established way of cost modulation in side-informed steganography, rounding "against" the rounding error is now penalized more. Thus, one can expect that this will have the biggest impact for harsh quantization (low quality JPEG). Also note that the sum of costs is now equal to the sum of the original costs

$$\rho_{ij}^{(\mathrm{TP})}(-1) + \rho_{ij}^{(\mathrm{TP})}(+1) = 2\rho_{ij}. \tag{4.3.3}$$

4.4 Experiments

This section contains the results of all experiments and their interpretation. We begin with SI-UNIWARD with the round quantizer and contrast the old (TD) cost modulation with the new one (TP). Then, we focus on "trunc" JPEGs and use the new rule for cost modulation in SI-UNIWARD (the old rule is inapplicable in this source).

4.4.1 Dataset

Our dataset was derived from 47,260 RAW images provided as part of the steganalysis competition ALASKA.[1] Available from the same web site is the script for developing the RAW images to the true-color (24 bit) TIFF format.[2] Then, the image was converted to grayscale, leaving the pixel values represented as "double," and resized using the cubic kernel so that the smaller side is 256, and finally centrally cropped to 256×256. The reader is referred to the above-cited ALASKA web site for more information the script. Pixel values were stored as integers before compression.

The database was randomly split into training, validation, and testing sets with 40,460, 3,200, and 3,600 images. Detectors trained as classifiers with rich models were trained on the union of the training and validation sets.

4.4.2 Evaluation metric

The detection performance was measured with the total classification error under equal priors on the test set

$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2}(P_{\mathrm{FA}} + P_{\mathrm{MD}}), \tag{4.4.1}$$

where P_{FA} and P_{MD} stand for the false-alarm and missed-detection probabilities.

4.4.3 Round JPEGs

Table 4.1 and Figures 4.1.1, 4.2.1 contrast the performance of ternary SI-UNIWARD as proposed in [31] (with TD modulation of costs) and the proposed version with costs modulated by minimum perturbation (TP). The tested payloads were 0.2, 0.4, 0.6, and 0.8 bits per non-zero AC DCT coefficient (bpnzac). While the impact of the new cost modulation is the largest for low quality factors and large payloads, improvement in security is observed for every tested scenario, with the exception of the largest quality factor 95 with payload 0.8 bpnzac, where the schemes attain the same level of detectability, and for quality factors 75 and 95 with payload 0.2 where the algorithms are virtually undetectable. In particular, with the selection-channel-aware (SCA) GFR feature set [119, 30], for quality 75, the improvement in security is almost 3% in terms of P_{E} for the largest payload. This gain decreases to 1.5–2% for quality 85. For the highest tested quality of 95, the improvement was less than 1%.

[1] https://alaska.utt.fr

[2] We modified the conversion script to only use the 'dem_amaze.pp3' RAW converter.

		QF 75				QF 85				QF 95			
Detector	Modulation	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
GFR	TD	**0.4796**	0.4071	0.2968	0.1822	0.4814	0.4147	0.3103	0.2011	**0.4838**	0.4517	0.3728	**0.2707**
	TP	0.4788	**0.4147**	**0.3203**	**0.2104**	**0.4828**	**0.4285**	**0.3269**	**0.2199**	0.4804	**0.4518**	**0.3831**	0.2704
SRNet	TD	**0.4658**	0.3337	0.1982	0.0953	0.4662	0.3381	0.1957	0.0984	0.4756	0.3664	0.2354	0.1198
	TP	0.4653	**0.3591**	**0.2182**	**0.1201**	**0.4713**	**0.3450**	**0.2186**	**0.1075**	**0.4808**	**0.3887**	**0.2467**	**0.1314**

Table 4.1: Detection error P_{E} of ternary SI-UNIWARD with (old) cost modulation by difference (TD) and (new) modulation by minimum perturbation (TP) with SCA-GFR / GFR feature set (whichever is better), ensemble classifier [97] and SCA-SRNet / SRNet (whichever is better).

The selection channel supplied to the SCA-SRNet was computed from the non-modulated costs because the modulation (side-information) is not available to the steganalyst. Interestingly, in most cases the selection channel actually hurts the performance of the SRNet. We conjecture that this may be due to the imprecise selection channel. The improvement in security offered by the new modulation is consistent with what was observed with rich models.

4.4.4 The trunc quantizer

Many portable imaging devices today use a slightly different implementation of JPEG compression that employs the operation of truncation for quantizing DCT coefficients [3, 14]. This quantizer essentially rounds towards zero instead of the nearest integer. Formally, the precover value x_{ij} is quantized to the nearest integer smaller than or equal to x_{ij} when $x_{ij} \geq 0$, and to the nearest integer larger than or equal to x_{ij} when $x_{ij} < 0$. This way of quantizing is adopted probably due to an easier hardware implementation.

Applying the original (TD) rule for modulation in this source leads to obvious problems because the rounding error $0 \leq e_{ij} < 1$ for $x_{ij} > 0$ and $-1 < e_{ij} \leq 0$ for $x_{ij} < 0$. A modulation factor $1 - 2|e_{ij}|$ would thus lead to negative costs. Moreover, it does not correspond to what one would intuitively expect because zero cost should be associated with $e_{ij} \approx 0$ or $|e_{ij}| \approx 1$. Additionally, precover values that are quantized to 0 experience $-1 < e_{ij} < 1$, while we require zero cost for $|e_{ij}| \approx 1$.

Computing the modulation factors as the minimum perturbation that makes the precover round to the desired stego value, for positive x_{ij}, $\rho_{ij}^{(\text{TP})}(+1) = (1 - e_{ij})\rho_{ij}$ and $\rho_{ij}^{(\text{TP})}(-1) = e_{ij}\rho_{ij}$, and for negative x_{ij}, $\rho_{ij}^{(\text{TP})}(+1) = -e_{ij}\rho_{ij}$ and $\rho_{ij}^{(\text{TP})}(-1) = (1 + e_{ij})\rho_{ij}$, which can be written in a more compact form :

$$\rho_{ij}^{(\text{TP})}(\text{sign}(e_{ij})) = (1 - |e_{ij}|)\rho_{ij} \tag{4.4.2}$$

$$\rho_{ij}^{(\text{TP})}(-\text{sign}(e_{ij})) = |e_{ij}|\rho_{ij}. \tag{4.4.3}$$

For x_{ij} such that $[x_{ij}] = 0$, the minimum perturbation is different due to the character of the quantizer :

$$\rho_{ij}^{(\text{TP})}(+1) = (1 - e_{ij})\rho_{ij} \tag{4.4.4}$$

$$\rho_{ij}^{(\text{TP})}(-1) = (1 + e_{ij})\rho_{ij}. \tag{4.4.5}$$

Note that for $[x_{ij}] \neq 0$, the sum of both costs is equal to ρ_{ij}, while for the zero bin $\{x \in \mathbb{R} \big| [x] = 0\}$ the sum is *twice* as large: $2\rho_{ij}$. This makes intuitive sense because the quantization bin for zero coefficients is two times larger than for any other bin, and it is crucial for SI-UNIWARD to work properly in trunc JPEGs [14]. Table 4.2 shows the performance of (TP) modulation in trunc JPEGs.

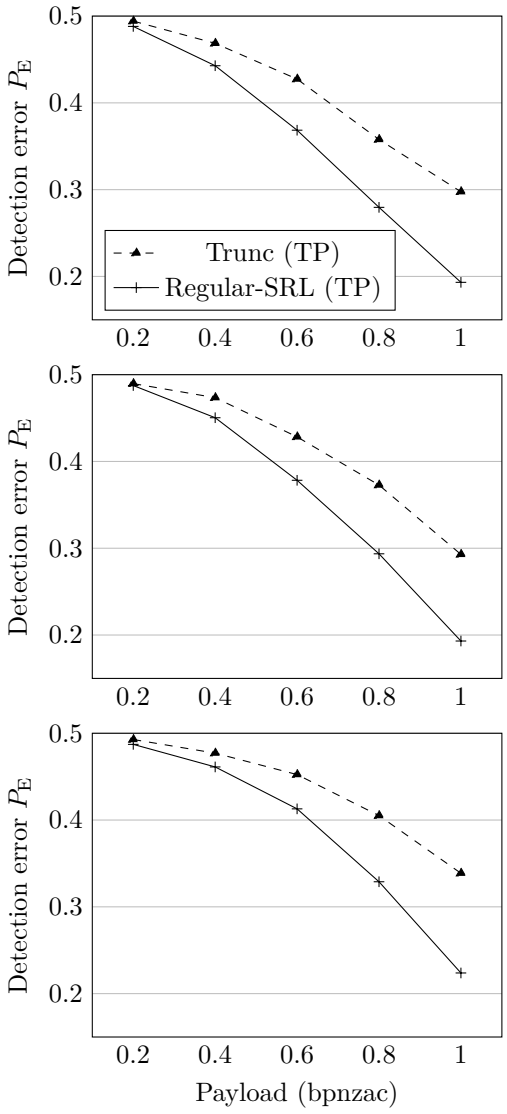

Figure 4.4.1: Detection error P_E of GFR for SI-UNIWARD with (new) modulation by minimum perturbation (TP) in trunc JPEGs (dashed) and with standard JPEGs with payload correction according to SRL (solid) at quality factors 75, 85, and 95.

To further validate the correctness of the (TP) cost modulation in trunc JPEGs, we compared the performance of SI-UNIWARD in regular JPEGs (with the round quantizer) embedded with payload size adjusted for constant statistical detectability according to the square root law (SRL) [89, 42, 87, 86] for a fair comparison. In particular, the relative payload in bpnzac in the round source, α_{round}, was scaled as

$$\alpha_{round} = \alpha_{trunc} \cdot \sqrt{\frac{N_{trunc}}{N_{round}}} \cdot \frac{\log(N_{round})}{\log(N_{trunc})}, \tag{4.4.6}$$

where N_{trunc} and N_{round} stand for the number of non-zero AC DCT coefficients from a given image in trunc and round JPEGs, respectively. Table 4.3 and Figure 4.4.1 show that even with the adjustment of the payload size according to the SRL, the (TP) cost modulation in trunc JPEGs is still more secure than in round JPEGs. This seems to indicate that the trunc source is harder to steganalyze.

	QF 75				QF 85				QF 95			
Detector	0.4	0.6	0.8	1	0.4	0.6	0.8	1	0.4	0.6	0.8	1
SCA-GFR	**0.4633**	**0.4256**	0.3631	0.2997	**0.4650**	**0.4147**	**0.3629**	**0.2919**	0.4868	0.4544	**0.4050**	**0.3261**
GFR	0.4686	0.4274	**0.3578**	**0.2976**	0.4735	0.4283	0.3728	0.2929	**0.4771**	**0.4524**	0.4051	0.3387
SRNet	0.4349	0.3499	0.2700	0.1908	0.4354	**0.3575**	0.2606	0.1839	**0.4561**	**0.3808**	**0.2876**	**0.1953**
SCA-SRNet	0.4349	**0.3475**	**0.2681**	**0.1840**	**0.4316**	0.3588	**0.2559**	**0.1756**	0.4569	0.3973	0.3140	0.2115

Table 4.2: Detection error P_{E} of ternary SI-UNIWARD with minimum perturbation (TP) in trunc JPEGs with SCA-GFR / GFR feature set, ensemble classifier and SCA-SRNet / SRNet.

	QF 75				QF 85				QF 95			
JPEGs	0.4	0.6	0.8	1	0.4	0.6	0.8	1	0.4	0.6	0.8	1
round SRL	0.4428	0.3685	0.2796	0.1932	0.4504	0.3783	0.2936	0.1931	0.4612	0.4129	0.3289	0.2238
trunc	**0.4686**	**0.4274**	**0.3578**	**0.2976**	**0.4735**	**0.4283**	**0.3728**	**0.2929**	**0.4771**	**0.4524**	**0.4051**	**0.3387**

Table 4.3: Detection error P_{E} of ternary SI-UNIWARD with minimum perturbation (TP) in trunc JPEGs and round JPEGs with payload scaled by SRL with GFR feature set, ensemble classifier.

4.5 Conclusions

Side-informed steganography is a term used for embedding with side-information, usually in the form of the unquantized cover called the precover. The quantization error e is used to adjust the costs of changing the cover element. In ternary schemes, this change can be either by $\text{sign}(e)$ or by $-\text{sign}(e)$, which can be interpreted as quantizing the precover to the second and third closest cover value, respectively. An established way to adjust the costs of both changes is to multiply the cost by $1-2|e|$ and by 1, respectively, which leads to unequal embedding change probabilities that prefer changing the cover element to the second closest value. This modulation is heuristically justified as the difference between the quantization errors to the corresponding stego and cover values.

In this work, we challenge this rule and propose modulation factors in the form of the minimal perturbation that needs to be applied to the precover to quantize to the desired stego value. Under this new rule, the modulation factor for the change by $\text{sign}(e)$ (to the second closest value) stays the same, $1-2|e|$, but it becomes $1+2|e|$ when quantizing to the third closest value, i. e., by $-\text{sign}(e)$. Penalizing such changes more has the biggest impact when the quantization is harsh, such as for low JPEG quality. In the spatial domain, where the quantization is fine, both rules give approximately the same performance.

For SI-UNIWARD in the JPEG domain, we observed an improvement by up to 3% in terms of P_{E} for quality 75 and the largest tested payloads (0.6 and 0.8 bpnzac). The gain generally diminishes with decreased payload and with increased JPEG quality. For quality 85 and 95, the largest gain was about 2% and 0.8%.

Our current work focuses on replacing the heuristics by deriving the embedding change rates from a precover model and the impact of embedding on the model similar to what was proposed in [33].

All code used to produce the results in this paper, including the network configuration files are available from `http://dde.binghamton.edu/download/`.

Chapter 5

Turning Cost-Based Steganography into Model-Based

Most modern steganographic schemes embed secrets by minimizing the total expected cost of modifications. However, costs are usually computed using heuristics and cannot be directly linked to statistical detectability. Moreover, as previously shown by Ker at al., cost-based schemes fundamentally minimize the wrong quantity that makes them more vulnerable to knowledgeable adversary aware of the embedding change rates. In this paper, we research the possibility to convert cost-based schemes to model-based ones by postulating that there exists payload size for which the change rates derived from costs coincide with change rates derived from some (not necessarily known) model. This allows us to find the steganographic Fisher information for each pixel (DCT coefficient), and embed other payload sizes by minimizing deflection. This rather simple measure indeed brings sometimes quite significant improvements in security especially with respect to steganalysis aware of the selection channel. Steganographic algorithms in both spatial and JPEG domains are studied with feature-based classifiers as well as CNNs.

5.1 Introduction

Steganography is another term for covert communication. Instead of communicating the actual message directly, or its encrypted form, it is hidden (embedded) in another cover object. Digital images are especially convenient covers for steganography because their individual elements (pixels or DCT coefficients in a JPEG file) can be slightly modified without changing the semantic meaning of the image. The main requirement here is that the stego objects carrying secrets should be statistically indistinguishable from cover objects [16]. Once the existence of a steganographic channel can be reliably established, the steganographic system is considered broken even if the adversary cannot read the secrets.

All modern steganographic schemes for images are content adaptive in the sense that they prefer modifying cover elements in complex or noisy parts of the image where it is more difficult for the adversary to detect the statistical impact of embedding changes [70, 98, 74, 115, 116, 66, 67]. Most stego schemes are "cost based" because each cover element $i \in \{1, \dots, N\}$ is assigned a cost $\rho_i \geq 0$ of changing its value. The required secret payload is then embedded so that each cover element is modified with probability β_i that minimizes the expected sum of costs of all changed pixels $d = \sum_{i=1}^{N} \beta_i \rho_i$, the embedding distortion. This problem is recognized as source coding with fidelity constraint [117] for which near optimal[1] coding has been devised [41]. In particular, when the

[1] In the sense of the corresponding rate–distortion bound.

embedding is allowed to change each cover element by ±1 with equal costs, the embedding change rates that minimize the expected distortion are

$$\beta_i = \frac{\exp(-\lambda\rho_i)}{1+2\exp(-\lambda\rho_i)}, \tag{5.1.1}$$

where the Lagrange multiplier $\lambda > 0$ is determined from the payload constraint (for the payload-limited sender)

$$\sum_{i=1}^{N} H_3(\beta_i) = m, \tag{5.1.2}$$

where m is the total number of bits to be embedded and $H_3(x) = -2\beta_i \log \beta_i - (1-2\beta_i)\log(1-2\beta_i)$ is the ternary entropy (payload) embedded at cover element i.

There are several issues with cost-based steganography. First of all, the costs themselves are usually computed using heuristic reasoning and cannot be easily related to statistical detectability of embedding changes. Second, this framework does not take into account a knowledgeable adversary aware of the embedding change rates β_i also known as the selection channel. In practice, this leads to embedding that is "overly adaptive," allowing the adversary to improve her detection accuracy using selection-channel-aware (SCA) features, such as [35, 123, 30, 32] or SCA convolutional neural networks (CNNs) [8, 133].

As shown in [**?**], considering steganography as a zero-sum game between the steganographer and the steganalyst, at equilibrium the sender should select β_i that minimize the statistical detectability, which is asymptotically directly linked to the so-called deflection coefficient

$$\delta^2 \propto \frac{1}{2}\sum_{i=1}^{N} \beta_i^2 I_i, \tag{5.1.3}$$

where I_i is the steganographic Fisher information [40, 85] at cover element i. In particular, the optimal change rates satisfy for each i

$$\beta_i I_i = H_3'(\beta_i), \tag{5.1.4}$$

where $H_3'(x)$ is the derivative of $H_3(x)$, subject to the same payload constraint. In practice, this is usually done by solving 5.1.4 and (5.1.2) numerically with a binary search over λ [56, 115, 116].

MiPOD [115] is an example of a steganographic scheme that minimizes the power of the most powerful detector an adversary can build when modeling the noise residuals in a digital image as independent realizations of zero-mean Gaussian random variables with variances σ_i^2 estimated for each cover element i. In this case, the steganographic Fisher information is $I_i \approx 1/\sigma_i^4$ in the fine quantization limit ($\sigma_i^2 > 1$).

Feature-Correction Method (FCM) [93], and approaches based on embedding while minimizing distance in some feature space, such as ASO [**?**], and Adv-Emb [**?**], are not truly model-based, because there is no underlying statistical model there, but are again distortion based with the measure of distortion computed as some distance in a selected feature space.

In this paper, we research the possibility to interpret cost-based embedding schemes as model-based schemes similar to MiPOD. We start with the assumption that, for some relative payload $\alpha = m/N$, the embedding change rates β_i computed from the costs as in (5.1.1) are the optimal change rates for some (unknown) cover model, derive the corresponding Fisher information, and then for all other payloads, we embed by minimizing the deflection (5.1.3). We expect the improvement in security to be especially noticeable for the case of a knowledgeable adversary who knows the embedding change rates β_i, i. e., when steganalyzing with SCA rich models or SCA versions of CNN detectors.

In the next section, we explain the main idea behind converting a cost-based scheme to a model based one. Section 5.3 contains the results of experiments with HILL and WOW. The improvement

in security is shown on two datasets with detectors built as rich models as well as deep CNNs. JPEG-domain schemes J-UNIWARD and UED-JC are studied experimentally in Section 5.4 for two quality factors 75 and 95. The reported gains are especially large for UED and for the smaller quality factor. Interpreting HILL's costs as reciprocals of local standard deviation estimates, in Section 5.5 we study a version of MiPOD with this different variance estimator. The paper is summarized in Section 7.5.

5.2 Costs to model

A brief inspection of the current literature on steganalysis in spatial domain (e. g., [8]) reveals that cost-based steganographic systems that do not use side-information at the sender, such as HILL [98], exhibit approximately the same level of empirical security as the model-based MiPOD [115]. Fundamentally, however, they are very different with HILL minimizing an objective function that is linear in change rates while MiPOD minimizes deflection, which is quadratic in change rates. Since practical embedding with the model-based MiPOD requires converting the optimal change rates determined by (5.1.4) to costs by inverting (5.1.1) and applying syndrome-trellis codes, one can interpret MiPOD as an embedding scheme with payload-dependent costs (also see Section 5, Fig. 2 in [56]). In this section, we explore this idea in reverse.

The formula for costs is usually derived heuristically through feedback provided by empirical steganalysis. For example, when designing HILL [98], the authors experimented with various sizes of the two low-pass filters. The authors of UNIWARD [74, 69] explored different wavelet bases and their supports as well as a range of values for the stabilizing constant [34]. And this is usually done for a fixed relative payload selected so that the detectability is not too small or too large to better see the impact of various design choices. In the spatial domain, the payload size of 0.4 bpp (bits per pixel) is a popular choice, also because it has been used in the steganalysis competition BOSS [4]. Thus, it is reasonable to assume that this empirical process leads to an embedding scheme that is *near optimal for the chosen payload and the dataset given the current status of steganalysis*. It has already been shown in [**?**] that steganography tends to be over-optimized for a given source of images. This is confirmed by the above observation that both HILL and MiPOD achieve a similar level of empirical detectability and the fact that no substantial improvement in additive steganography has been reported in the past six years of rather intense research.

Thus, we make an assumption that, given some embedding scheme with costs ρ_i, there exists a relative payload α_D (bpp), which we call the *design payload*, for which the embedding change rates $\beta_i^{(\alpha_D)}$ derived from the costs are near optimal for the current status of steganalysis. Then, we derive the corresponding Fisher information for each pixel, $I_i^{(\alpha_D)}$, so that the deflection $\delta^2 = \frac{1}{2}\sum_{i=1}^N \beta_i^2 I_i^{(\alpha_D)}$ achieves its minimum value when $\beta_i = \beta_i^{(\alpha_D)}$ under the same payload constraint. Using the method of Lagrange multipliers, it can be easily shown that this happens exactly when

$$I_i^{(\alpha_D)} = \frac{\rho_i}{\beta_i^{(\alpha_D)}}. \tag{5.2.1}$$

Having determined the Fisher information for each pixel, we can now embed other payload sizes $\alpha \neq \alpha_D$ by minimizing the deflection

$$\delta^2(\alpha) = \frac{1}{2}\sum_{i=1}^N \beta_i^2 I_i^{(\alpha_D)} \tag{5.2.2}$$

subject to $\sum_{i=1}^N H_3(\beta_i) = \alpha N$. A graphical representation of above protocol is shown in Figure 5.2.1.

Note that this approach does not inform us about the model that is responsible for the steganographic Fisher information. We merely determine I_i, which could correspond to many different models.

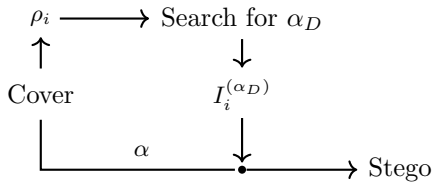

Figure 5.2.1: Embedding relative message α bpp (bpnzac) with design payload α_D for arbitrary cost-based steganographic scheme. Notice that the costs ρ_i are used only to compute the Fisher Information for each pixel $I_i^{(\alpha_D)}$.

HILL (SRM)						
$\alpha_D \backslash \alpha$	0.05	0.1	0.2	0.3	0.4	0.5
0.05	**0.4739**	**0.4416**	0.3636	0.2951	0.2454	0.2017
0.1	0.4712	0.4364	**0.3735**	0.3065	0.2503	0.1994
0.2	0.4643	0.4336	0.3669	**0.3106**	0.2525	0.2097
0.3	0.4587	0.4303	0.3639	0.3056	**0.2537**	0.2067
0.4	0.4544	0.4206	0.3666	0.3067	0.2525	**0.2115**
0.5	0.4548	0.4127	0.3481	0.3005	0.2475	0.2077

HILL (maxSR				
$\alpha_D \backslash \alpha$	0.05	0.1	0.2	0
0.05	0.4307	0.3732	0.2916	0.2
0.1	0.4452	0.3909	0.3067	0.2
0.2	0.4457	0.4024	0.3189	0.2
0.3	0.4484	**0.4056**	0.3282	0.2
0.4	**0.4502**	0.4025	0.3327	0.2
0.5	0.4440	0.4031	**0.3353**	**0.2**

Table 5.1: Detection error P_{E} of model-based HILL for different design payloads α_D and embedded payloads α. Left: SRM, Right: maxSRMd2, ensemble classifier, BOSSbase. Regular HILL corresponds to the diagonal ($\alpha_D = \alpha$).

	α	0.05	0.1	0.2	0.3	0.4
Regular	SRNet	0.3893	0.3192	0.2325	0.1779	0.1465
HILL	SCA-SRNet	0.3992	0.3164	0.2167	0.1717	0.1360
MB-HILL	SRNet	**0.4188**	**0.3468**	0.2449	0.1811	0.1444
$\alpha_D = 0.5$	SCA-SRNet	0.4751	0.3591	**0.2387**	**0.1777**	**0.1393**

Table 5.2: Detection error P_{E} of SRNet and SCA-SRNET for HILL and model-based HILL ($\alpha_D = 0.5$ bpp) in downsampled BOSSbase + BOWS2.

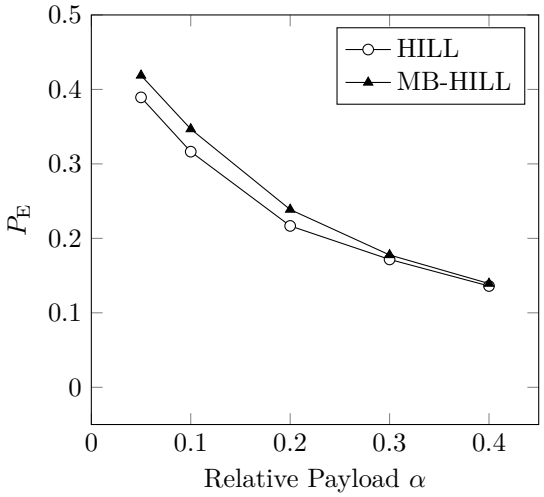

Figure 5.2.2: Detection error P_{E} of the best detector (SRNet or SCA-SRNet) for HILL and model-based HILL ($\alpha_D = 0.5$ bpp) in downsampled BOSSbase + BOWS2.

5.3 Spatial domain

In this section, we focus on spatial-domain steganographic algorithms HILL [98] and WOW [70]. Since both have been designed on the standard dataset BOSSbase 1.01 [4] containing 10,000 512×512 grayscale images, we search for the best design payload α_D on the same dataset unless mentioned otherwise. The FLD ensemble [97] with the spatial rich model (SRM) [54] and maxSRM [35] was trained on 5,000 randomly selected images and tested on the remaining 5,000.

5.3.1 Model-based HILL

Table 5.1 shows the results for HILL in terms of P_{E}, the total classification error under equal priors for the cover and stego classes. The boldface font highlights the most secure algorithm version, which is to be compared with the diagonal ($\alpha = \alpha_D$) corresponding to regular HILL. Note that the results are vastly different depending on the steganalysis features. For SRM, which is an ignorant adversary (one who does not use the knowledge of the selection channel), there is no clear design payload that would always give the best results. Also, the impact on security is quite small. In contrast, detection with a knowledgeable adversary (maxSRMd2) indicates that the best overall design payload is $\alpha_D = 0.5$ bpp (for the two smallest tested payloads the differences between $\alpha_D = 0.3, 0.4$, and 0.5 are small). The largest boost in empirical security is 1.7% for payload $\alpha = 0.2$.

We repeated the same experiment with the CNN SRNet and its SCA version [8]. Because large CNNs, such as the SRNet, cannot be trained on 512×512 images on GPUs with 12 GB memory with a reasonable batch size, we used the union of BOSSbase and BOWS2 whose images were downsampled to 256×256 pixels using Matlab's imresize with default parameters. As in [8, 133], this 20,000 image dataset was split into 14,000 (10,000 BOWS2 and 4,000 randomly chosen from BOSSbase) for training, 1,000 BOSSbase images for validation, and 5,000 for testing.

Technically, the design payload should be searched for anew for this dataset and detector. Due to the much more computationally demanding training of the SRNet, however, we only compare model-based HILL for $\alpha_D = 0.5$ and regular HILL (Table 5.2). Comparing the best detector (SRNet vs. SCA-SRNet[2]) for each embedding algorithm in Figure 5.2.2, we observe an empirical gain in security ranging from almost 3% for the smallest payloads to almost no gain for $\alpha = 0.4$.

5.3.2 Model-based WOW

Searching for the best design payload on BOSSbase with maxSRMd2 and the ensemble classifier, it also appears to be close to $\alpha_D = 0.5$ bpp. Since WOW is known to be overly content-adaptive in the sense that its security decreases significantly with selection-channel-aware attacks, the impact of making it model-based is larger than for HILL. The detection error P_{E} shown in Figure 5.3.1 is about 4% larger for the two smallest payloads for model-based WOW than for the original cost-based algorithm.

On the dataset of downsampled images, based on our investigation with maxSRMd2, the best design payload is larger, $\alpha_D = 0.7$ bpp. In Figure 5.3.2, we contrast the detection error of SRNet on model-based WOW and WOW ranges from 3.4% for the smallest payload of 0.05 bpp to 0.7% for 0.2 bpp. The empirical security of both algorithms appears similar for the two largest payloads. The actual values of the detection error appear in Table 5.4 at the end of this paper.

5.4 JPEG domain

In the JPEG domain, we investigated the embedding algorithms J-UNIWARD and UED-JC [67]. For the database of larger 512×512 images, we steganalyzed with selection-channel-aware Gabor

[2] In some cases, SCA-SRNet performs worse than SRNet.

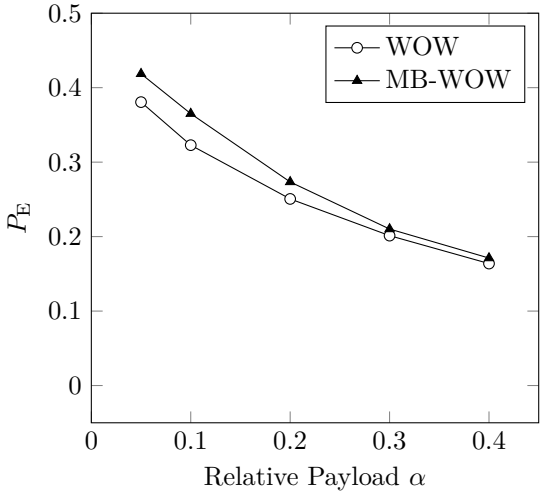

Figure 5.3.1: Detection error P_{E} of maxSRMd2 for WOW and model-based WOW ($\alpha_D = 0.5$ bpp) in BOSSbase.

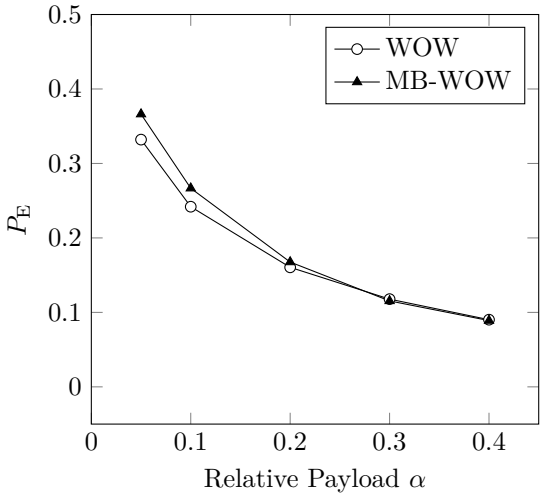

Figure 5.3.2: Detection error P_{E} of the best detector (SRNet or SCA-SRNet) for WOW and model-based WOW ($\alpha_D = 0.7$ bpp) in downsampled images BOSSbase + BOWS2.

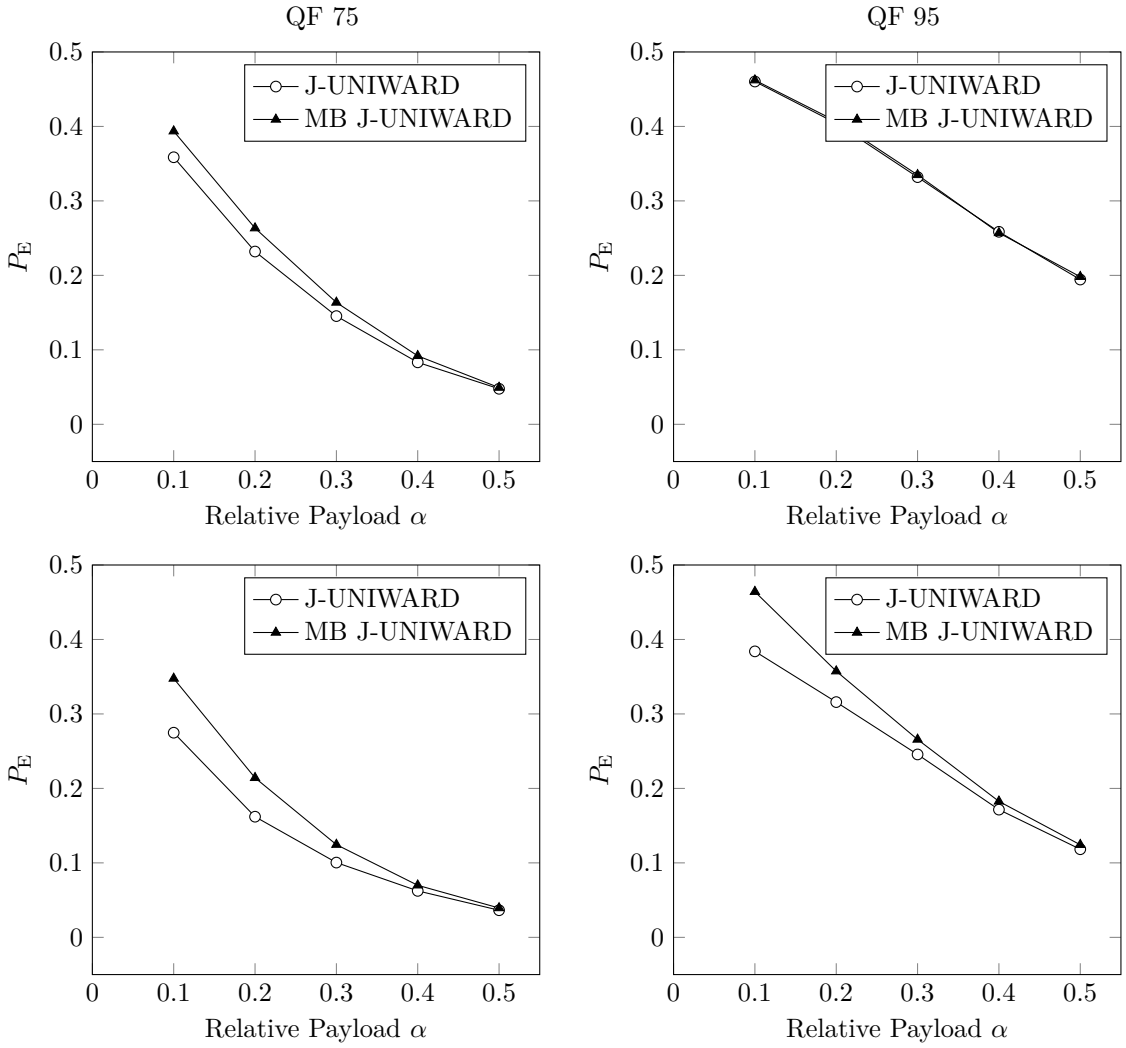

Figure 5.4.1: Detection error P_E for J-UNIWARD and model-based J-UNIWARD ($\alpha_D = 0.6$ bpnzac) when steganalyzing with SCA-GFR on BOSSbase (top) and with SCA-SRNet (or SRNet, whichever is better) on downsampled BOSSbase + BOWS2 (bottom) for quality 75 and 95.

Phase Aware Residuals, SCA-GFR [119, 30], while, as above, the SRNet and SCA-SRNet were used on the database of downsampled images. The split of the datasets was the same as for the experiments in the spatial domain.

5.4.1 J-UNIWARD

For J-UNIWARD, the results are graphically displayed in Figure 5.4.1 showing the detection error of J-UNIWARD and its model-based version with $\alpha_D = 0.6$ bpnzac. The gain in security is generally much larger than what was observed in the spatial domain. Also, it is larger for quality factor 75 than for 95. As before, the gain increases with decreasing payload. In particular, for quality 75 the gain was up to 3.5% with SCA-GFR and 7.3% with SCA-SRNet. While we observed almost no gain for quality 95 with SCA-GFR, the better detector (SCA-SRNet) showed more than 8% of improvement for the smallest payload.

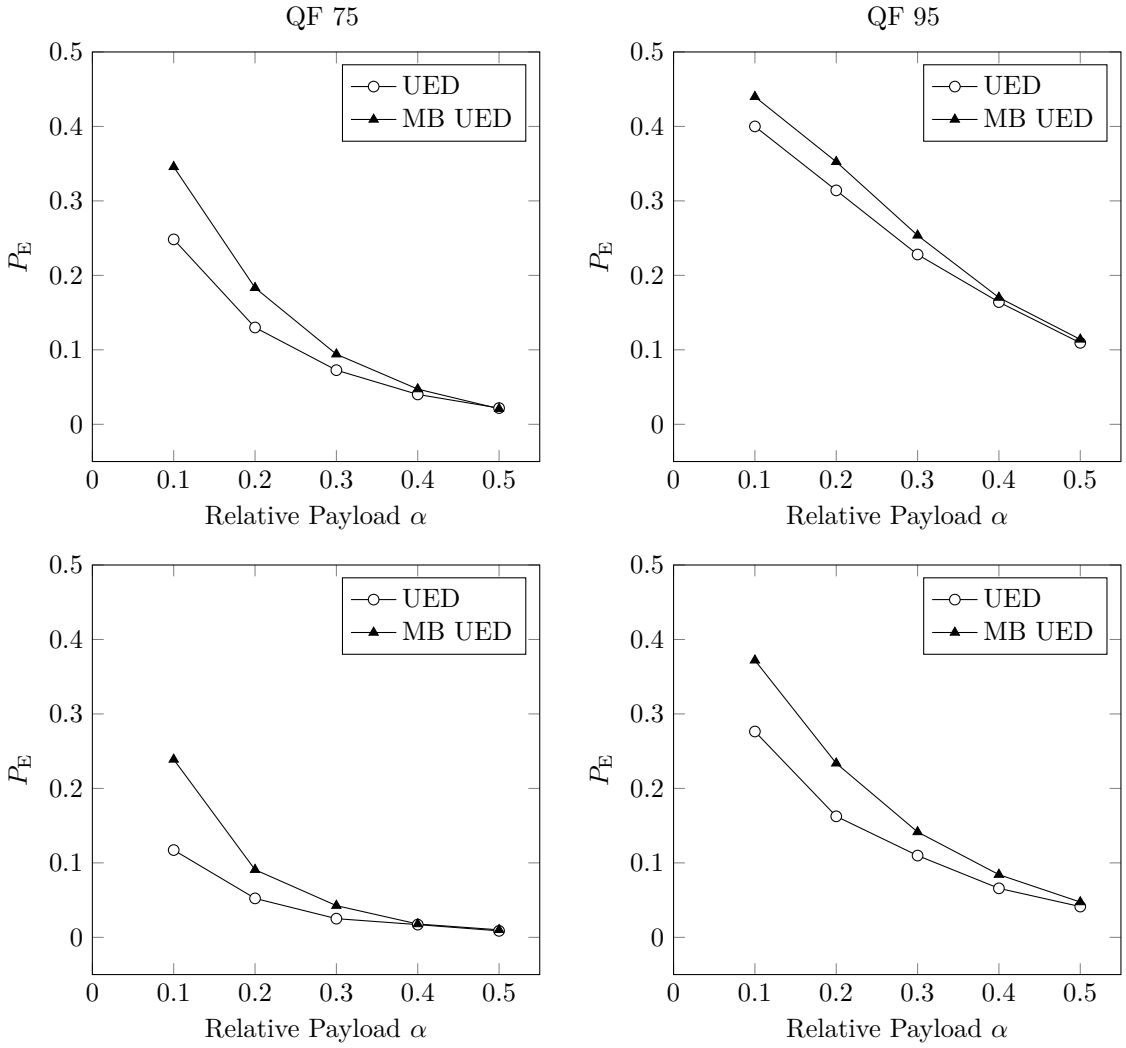

Figure 5.4.2: Detection error P_E for UED-JC and model-based UED-JC ($\alpha_D = 0.6$ bpnzac) when steganalyzing with SCA-GFR on BOSSbase (top) and with SCA-SRNet (or SRNet, whichever is better) on downsampled BOSSbase + BOWS2 (bottom) for quality 75 and 95.

5.4.2 UED

The embedding algorithm UED-JC benefits from our approach by far the most out of all tested stego methods in any domain. Figure 5.4.2 shows the detection error achieved on BOSSbase with SCA-GFR and on the downsampled images with (SCA)-SRNet for two quality factors. The gain is again larger on downsampled images when detecting with (SCA)-SRNet and is over 12% for the smallest payload. On BOSSbase with SCA-GFR, the gain on the smallest payload is about 10%. In both datasets, the gain diminishes to zero as α approaches α_D.

The actual values of the detection error from the graphs for J-UNIWARD and UED-JC appear in Table 5.4 at the end of this paper.

5.5 Interpreting HILL's costs

The main contribution of this paper is the realization that there is a cover model behind cost-based schemes and a method for estimating the model, its Fisher information. In this section, we take a closer look at the embedding algorithm HILL, and interpret its costs as reciprocal estimates of the local standard deviation. Equipped with this insight, we implement a model-based version of HILL with a Gaussian model of pixel residual, which is essentially a version of MiPOD with a different variance estimator.

HILL (**H**igh-pass, **L**ow-pass, **L**ow-pass) computes costs heuristically using a series of filtering operations. First, the 3×3 high-pass KB filter [88] F_{KB} is applied to the cover image $\mathbf{X}$, producing the KB residual $\mathbf{R} = \mathbf{X} \star F_{\text{KB}}$. Next, the absolute value of the KB residual is smoothed with a 3×3 averaging filter $A_{3\times3}$: $|\mathbf{R}| \star A_{3\times3}$. Finally, the reciprocal of this signal is smoothed by applying a 15×15 averaging filter $A_{15\times15}$:

$$\boldsymbol{\rho} = A_{15\times15} \star \frac{1}{|\mathbf{R}| \star A_{3\times3}}. \tag{5.5.1}$$

Ignoring the second low-pass filtering in Equation 5.5.1 for simplicity, the costs can be seen as reciprocal expectation of the absolute value of the KB residual $\rho_i \simeq 1/E[|R_i|]$ or a reciprocal of the Mean Absolute Deviation (MAD), assuming the KB residual is zero mean. Similar to the standard deviation (std), MAD is a description of a statistical spread of a random variable X. For a wide range of distributions typically used in image modeling (e. g., for the generalized Gaussian distribution and the generalized Gamma distribution), the expectation of absolute value is proportional to the standard deviation when fixing the remaining parameters, $E[|X|] \propto \sigma$. Thus, the reciprocal cost

$$\frac{1}{\rho_i} \simeq E[|R_i|] \propto \sigma_i. \tag{5.5.2}$$

This tells us that that HILL's costs can loosely be viewed as reciprocals of estimates of local standard deviation. Assuming the KB residual is locally Gaussian $R_i \sim \mathcal{N}(0, \sigma_i^2)$, the costs inform us about the standard deviations σ_i:

$$1/\rho_i \simeq E[|R_i|] = \sigma_i \sqrt{\frac{2}{\pi}}. \tag{5.5.3}$$

Note that, with a locally Gaussian residual model, we arrived at a different version of MiPOD with the following "HILL-inspired" plug-in variance estimator

$$\sigma_i^2 = \frac{\pi}{2\rho_i^2}. \tag{5.5.4}$$

Before subjecting this embedding scheme to practical tests, we first validate the model in the following fashion. Given image $\mathbf{X}$ with KB residual $\mathbf{R}$, we first estimate its local variance from HILL's costs (5.5.4) and then using MiPOD's variance estimator, respectively.[3] Then, we sample M times the multivariate Gaussian $(\widetilde{R}_1, \dots, \widetilde{R}_N)$, $\widetilde{R}_i \sim \mathcal{N}(0, \sigma_i^2)$, where N is the number of pixels in the image. Given these $M \times N$ random samples $\widetilde{\mathbf{R}}$, we compute their empirical probability mass function (histogram with 100 uniform bins) $h_{\widetilde{\mathbf{R}}}$ and compare it with the histogram $h_{\mathbf{R}}$ of the KB residual $\mathbf{R}$ using the discrete Kullback–Leibler divergence $D_{\text{KL}}(h_{\mathbf{R}}||h_{\widetilde{\mathbf{R}}})$. Executing this for 5,000 512×512 grayscale images $\mathbf{X}$ from the training subset of BOSSbase 1.01, in Figure 5.5.1 we show the box plot of the KL divergence across all 5,000 images obtained using both variance estimators. Note that if

[3] For MiPOD, this was achieved by passing the KB residual instead of the noise residual computed using the 2×2 Wiener filter to the parametric denoising algorithm (see Sec. V in [115]).

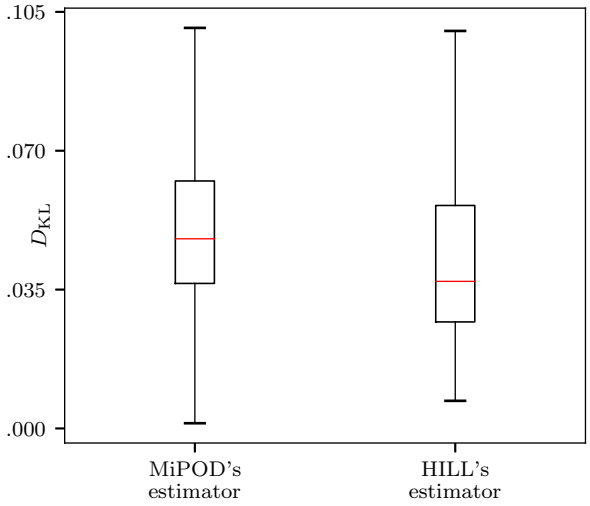

Figure 5.5.1: $D_{\mathrm{KL}}(R||\widetilde{R})$ with the KB residual variance estimated using HILL's costs and MiPOD's variance estimator. The red line shows the median, the bottom and top edges of the box indicate the 25th and 75th percentiles, and the whiskers length set to 1.5. Samples computed from 5,000 512×512 grayscale images from BOSSbase.

Variance		0.1	0.2	0.3	0.4
Eq. (5.5.4)	maxSRMd2	**0.3937**	**0.3206**	**0.2678**	**0.2213**
MiPOD	maxSRMd2	0.3800	0.3101	0.2552	0.2142
Eq. (5.5.4)	SRNet	**0.3390**	0.2470	0.1870	0.1545
	SCA-SRNet	0.3575	**0.2354**	**0.1826**	**0.1420**
MiPOD	SRNet	0.3213	0.2222	0.1553	0.1146
	SCA-SRNet	0.2952	0.1961	0.1384	0.1106

Table 5.3: Detection error P_{E} for MiPOD with variance estimator (5.5.4) and the original MiPOD estimator in BOSSbase (maxSRMd2 + ensemble) and in downsampled BOSSbase + BOWS2 (with (SCA)-SRNet).

the adopted and estimated model perfectly fit the KB residual, we would see a KL divergence near zero. The figure shows that using HILL's costs to estimate the KB variance is slightly better in terms of preserving the overall residual distribution.

Based on this observation, we implemented MiPOD with HILL's variance estimator (5.5.4). In order to focus on the effect of the variance estimator, we skip the Fisher Information smoothing step in MiPOD. Table 5.3 shows that the HILL-inspired estimator (5.5.4) provides better security than the original variance estimator in MiPOD, in agreement with the model validation shown in Figure 5.5.1.

5.6 Conclusions

Most steganographic schemes today are content adaptive, designed around the paradigm of minimizing the total embedding cost. Costs are, however, typically designed using intuitive heuristic rules, making it difficult, if possible at all, to link the impact of embedding to statistical detectability. Moreover, at least asymptotically for small payloads, the statistical detectability is quadratic in embedding change rates while the embedding distortion is linear. On the other hand, given the

success of cost-based steganography to avoid steganalysis, the costs must have some relationship to detectability.

Costs are typically designed from feedback provided by steganalysis on a selected dataset and usually for a fixed payload. In this paper, we postulate that there exists a relative payload for which the embedding change rates correspond to minimal statistical detectability for some unknown model of pixels (DCTs). For this so-called design payload, we convert the costs to the steganographic Fisher information. Although the underlying model is not known, with the Fisher information, we can embed other payloads with a model-based scheme by minimizing the deflection. As shown in this paper, this rather simple idea indeed leads to improved security, especially with respect to selection-channel-aware steganalysis. The gain typically increases with decreased payload. In JPEG domain, we observed larger gains for smaller quality factors than for large qualities. The gains for JPEG-domain algorithms are also generally larger than for spatial domain. The largest observed gains exceed 12% in terms of the total detection error under equal priors P_{E} for UED-JC at quality 75.

Inspired by the success of this simple idea, we also explore a model-based scheme, a version of MiPOD, with a different pixel variance estimator obtained by interpreting HILL's costs as reciprocal estimates of standard deviation from the KB residual. This algorithm indeed performs better than when estimating the variance of the KB residual with MiPOD.

QF	Steganography	Detector	Payload (bpp / bpnzac) 0.05	0.1	0.2	0.3	0.4	0.5
-	Regular WOW	SRM	0.4606	0.4113	0.3218	0.2563	0.2142	-
		maxSRMd2	0.3806	0.3228	0.2506	0.2013	0.1638	-
	MB WOW $\alpha_D = 0.5$	SRM	0.4515	0.3984	0.3284	0.2562	0.2078	-
		maxSRMd2	**0.4186**	**0.3651**	**0.2734**	**0.2102**	**0.1712**	-
	Regular WOW	SRNet	0.3415	0.2587	0.1701	0.1287	0.1010	-
		SCA-SRNet	0.3320	0.2419	0.1605	**0.1178**	**0.0902**	-
	MB WOW $\alpha_D = 0.7$	SRNet	**0.3662**	0.2678	0.1696	0.1208	0.0913	-
		SCA-SRNet	0.3766	**0.2667**	**0.1676**	0.1154	0.0890	-
75	J-UNIWARD	SRNet	-	0.3161	0.1931	0.1121	0.0707	0.0375
		SCA-SRNet	-	0.2748	0.1620	0.1004	0.0624	0.0364
	MB J-UNIWARD $\alpha_D = 0.6$	SRNet	-	0.3612	0.2196	0.1300	0.0814	0.0465
		SCA-SRNet	-	**0.3476**	**0.2142**	**0.1245**	**0.0699**	**0.0394**
	J-UNIWARD	SCA-GFR	-	0.3586	0.2320	0.1453	0.0832	0.0477
	MB J-UNIWARD $\alpha_D = 0.6$	SCA-GFR	-	**0.3936**	**0.2634**	**0.1636**	**0.0919**	**0.0493**
95	J-UNIWARD	SRNet	-	0.4418	0.3436	0.2594	0.1847	0.1306
		SCA-SRNet	-	0.3840	0.3159	0.2456	0.1715	0.1183
	MB J-UNIWARD $\alpha_D = 0.6$	SRNet	-	0.4772	0.3683	0.2694	0.1859	**0.1243**
		SCA-SRNet	-	**0.4641**	**0.3574**	**0.2657**	**0.1826**	0.1264
	J-UNIWARD	SCA-GFR	-	0.4603	0.4042	0.3319	**0.2585**	0.1944
	MB J-UNIWARD $\alpha_D = 0.6$	SCA-GFR	-	**0.4621**	**0.4069**	**0.3349**	0.2570	**0.1981**
75	UED	SRNet	-	0.1344	0.0571	0.0311	0.0196	0.0111
		SCA-SRNet	-	0.1172	0.0523	0.0251	0.0171	0.0087
	MB UED $\alpha_D = 0.6$	SRNet	-	**0.2389**	0.1003	0.0466	0.0224	**0.0101**
		SCA-SRNet	-	0.2419	**0.0908**	**0.0426**	**0.0179**	0.0126
	UED	SCA-GFR	-	0.2483	0.1300	0.0727	**0.0401**	**0.0218**
	MB UED $\alpha_D = 0.6$	SCA-GFR	-	**0.3457**	**0.1833**	**0.0941**	**0.0473**	0.0209
95	UED	SRNet	-	0.2966	0.1997	0.1253	0.0818	0.0534
		SCA-SRNet	-	0.2764	0.1725	0.1098	0.0658	0.0413
	MB UED $\alpha_D = 0.6$	SRNet	-	0.4036	0.2669	0.1696	0.1113	0.0625
		SCA-SRNet	-	**0.3720**	**0.2337**	**0.1415**	**0.0842**	**0.0474**
	UED	SCA-GFR	-	0.4000	0.3141	0.2280	**0.1641**	0.1094
	MB UED $\alpha_D = 0.6$	SCA-GFR	-	**0.4398**	**0.3525**	**0.2537**	**0.1702**	**0.1140**

Table 5.4: For completeness, this table shows the actual numerical values of the detection error P_{E} for all experiments in the main body of the paper that are reported only in a graphical form. All results with rich models are on BOSSbase 512×512 images with ensemble classifier as the detector. SRNet results are always on the union BOSSbase + BOWS2 downsampled to 256×256. For the JPEG domain, the smallest studied payload is 0.1 bpnzac.

Chapter 6

Reverse JPEG Compatibility Attack

A novel steganalysis method for JPEG images is introduced that is universal in the sense that it reliably detects any type of steganography as well as small payloads. It is limited to quality factors 99 and 100. The detection statistic is formed from the rounding errors in the spatial domain after decompressing the JPEG image. The attack works whenever, during compression, the discrete cosine transform is applied to integer-valued signal. Reminiscent of the well-established JPEG compatibility steganalysis, we call the new approach the "reverse JPEG compatibility attack." While the attack is introduced and analyzed under simplifying assumptions using reasoning based on statistical signal detection, the best detection in practice is obtained with machine learning tools. Experiments on diverse datasets of both grayscale and color images, five steganographic schemes, and with a variety of JPEG compressors demonstrate the universality and applicability of this steganalysis method in practice.

6.1 Introduction

The term "compatibility attack" is loosely used to describe a certain type of steganalysis detectors that identify stego objects by verifying either hard or probabilistic constraints that must be satisfied by all cover objects from a certain source. Typically, such attacks are universal in the sense that they work reliably on most steganographic methods as well as for very small payloads.

The first example of such an attack was the JPEG compatibility steganalysis [48] applicable whenever spatial-domain steganography is used to embed a secret in a decompressed JPEG cover image. The stego image will still bear strong traces of the JPEG compression, allowing an attacker to estimate the quantization matrix of the JPEG cover image. Since JPEG compression with a low quality factor is a many-to-one mapping, one could either mathematically prove or at least find overwhelming statistical evidence that a given 8×8 block of pixels with steganographic modifications cannot be obtained by decompressing any 8×8 block of quantized Discrete Cosine Transform (DCT) coefficients with the estimated quantization matrix. To make this attack less susceptible to loss of accuracy due to differences between JPEG compressors in practice, alternative versions of this idea were proposed by employing feature based machine learning detectors [103, 95]. Another version of this attack deals with steganalysis of LSB replacement [6, 7].

A different type of compatibility attack for color images was described in [63], where the authors show that mere eight bins in the co-occurrence corresponding to the 'minmax41c' submodel of the Color Rich Model (CRM) [62] hold all the detection power when the cover images are developed in 'dcraw' using AHD and PPG demosaicking algorithms. These eight bins are "violator bins"

that are nearly empty in cover images (this is the compatibility constraint) but get populated by steganography allowing thus construction of extremely accurate detectors.

A powerful compatibility constraint in the co-occurrence corresponding to the KB residual (SQUARE3x3 submodel) in the Spatial Rich Model (SRM) [54] was also identified in parity-aware version of the SRM in [55] for steganalysis of LSB replacement for cover sources with suppressed noise, such as decompressed JPEGs or filtered images.

The compatibility attack described in this paper only applies to JPEG images compressed with standard quantization matrices with quality 99 and 100. However, after reading Section 6.3.5, it should be clear to the reader that this attack will work for custom quantization matrices that can loosely be described as being "close" to 99 or 100. While this may seem as a severe limitation, based on the study conducted by the creators of the recent ALASKA steganalysis competition,[1] 14% of JPEG images with standard quantization matrices uploaded to Flickr have quality 100 and an additional 4% quality 99. This popularity of high quality factors may be due to the rapid decrease of storage prices combined with increased preference of users to preserve the quality of imagery they share on social platforms. Steganographers may also intentionally opt for larger JPEG qualities to increase the embedding capacity since many freely available steganographic programs, such as Jsteg [126], OutGuess [111], F5 [128], Steghide [68], Model Based Steganography [114], and JP Hide&Seek only embed in non-zero DCT coefficients. There also appears an increased interest within the forensics community in studying quantization noise during recompression with high quality factors [**?**, **?**].

After introducing notation, the basics of JPEG compression, and a few preliminaries in the next section, we explain the main idea behind the reverse JPEG compatibility attack in Section 6.3 by analyzing the rounding errors in the spatial domain after decompressing a JPEG image. A statistical hypothesis formulation of the detection problem allows studying the limitations of the attack. In Section 6.4, we describe three machine learning built detectors trained on rounding errors and identify the most accurate detector, which is further tested in Section 6.5 and Section 6.6 for universality and robustness to various implementations of JPEG compression, grayscale as well as color JPEGs, in established datasets, such as the union of BOSSbase and BOWS2, and a more realistic setting on the ALASKA dataset. After discussing countermeasures steganographers could use to improve the security for quality 100 in Section 6.7, the paper is summarized in Section 7.5.

6.2 Preliminaries

Boldface symbols are reserved for matrices and vectors. The symbol $'\cdot'$ is used to denote elementwise product between vectors / matrices of the same dimensions. Uniform distribution on the interval $[a, b]$ will be denoted $\mathcal{U}[a, b]$ while $\mathcal{N}(\mu, \sigma^2)$ is used for the Gaussian distribution with mean μ and variance σ^2. The operation of rounding x to an integer is the square bracket $[x]$. The set of all integers will be denoted $\mathbb{Z}$. The symbol $\triangleq$ is used whenever a new concept is defined.

6.2.1 Folded Gaussian distribution

For $X \sim \mathcal{N}(\mu, s)$ with $\mu \in \mathbb{Z}$, the rounding error $X - [X] \sim \nu(x; s)$, $-1/2 \le x < 1/2$, where

$$\nu(x; s) = \frac{1}{\sqrt{2\pi s}} \sum_{n \in \mathbb{Z}} \exp\left(-\frac{(x+n)^2}{2s}\right) \tag{6.2.1}$$

$$= \frac{1}{\sqrt{2\pi s}} \Bigg(e^{-\frac{x^2}{2s}} + e^{-\frac{(x-1)^2}{2s}} + e^{-\frac{(x+1)^2}{2s}} + \cdots + e^{-\frac{(x-n_0)^2}{2s}} + e^{-\frac{(x+n_0)^2}{2s}} + R(x; n_0, s) \Bigg) \tag{6.2.2}$$

[1] https://alaska.utt.fr

is the Gaussian distribution "folded" to the half-open interval $[-1/2, 1/2)$. The probability of specific rounding error $e \in [-1/2, 1/2)$ is basically a sum of the Gaussian distributions with means $\ldots -1+e, e, 1+e, \ldots$.

It is routine to show that the remainder $R(x; n_0, s)$ is smaller than the n_0th term divided by $e^{n_0/s}-1$ for all $x \in [-1/2, 1/2)$:

$$\begin{aligned} R(x; n_0, s) &\leq \left(e^{-\frac{(x-n_0)^2}{2s}} + e^{-\frac{(x+n_0)^2}{2s}} \right) \frac{1}{e^{n_0/s}-1} \\ &\leq \frac{2e^{-\frac{(n_0-1/2)^2}{2s}}}{e^{n_0/s}-1} \triangleq R_{\max}(n_0, s). \end{aligned} \tag{6.2.3}$$

Thus, the truncated sum

$$\nu(x; s, n_0) \triangleq \frac{1}{\sqrt{2\pi s}} \sum_{n=-n_0}^{n_0} \exp\left(-\frac{(x+n)^2}{2s} \right) \tag{6.2.4}$$

approximates $\nu(x; s)$

$$\max_{x \in [-1/2, 1/2)} |v(x; s) - \nu(x; s, n_0)| < \delta, \tag{6.2.5}$$

once n_0 becomes large enough to satisfy

$$\frac{1}{\sqrt{2\pi s}} R_{\max}(n_0, s) < \delta. \tag{6.2.6}$$

6.2.2 Basics of JPEG compression

JPEG compression proceeds by dividing the image into 8×8 blocks, applying the DCT to each block, dividing the DCT coefficients by quantization steps, and rounding to integers. The coefficients are then arranged in a zig-zag fashion and losslessly compressed to be written as a bitstream into the JPEG file together with a header. We first describe this process for a grayscale image.

For better readability, everywhere in this paper, i, j will be strictly used to index pixels and k, l will index DCT coefficients. The original uncompressed 8-bit grayscale image with $N_1 \times N_2$ pixels is denoted $\mathbf{x} \in \{0, 1, \ldots, 255\}^{N_1 \times N_2}$. Constraining $\mathbf{x} = (x_{ij})$ to one specific 8×8 block, we will use indices $0 \leq i, j \leq 7$ to index the pixels in this block. During JPEG compression, the DCT coefficients before quantization, $d_{kl} \in \mathbb{R}$, are obtained using the formula $d_{kl} = \mathrm{DCT}_{kl}(\mathbf{x}) \triangleq \sum_{i,j=0}^{7} f_{kl}^{ij} x_{ij}$, $0 \leq k, l \leq 7$, where

$$f_{kl}^{ij} = \frac{w_k w_l}{4} \cos\frac{\pi k(2i+1)}{16} \cos\frac{\pi l(2j+1)}{16}, \tag{6.2.7}$$

$w_0 = 1/\sqrt{2}$, $w_k = 1$ for $0 < k \leq 7$ are the discrete cosines. Before applying the DCT, each pixel is adjusted by subtracting 128 from it during JPEG compression, a step we omit here since it has no effect on our analysis.

The quantized DCTs are $c_{kl} = [d_{kl}/q_{kl}]$, $c_{kl} \in \{-1024, \ldots, 1023\}$, where q_{kl} are quantization steps in a luminance quantization matrix, which is supplied in the header of the JPEG file.

During decompression, the above steps are reversed. For a block of quantized DCT coefficients c_{kl}, the corresponding block of non-rounded pixel values after decompression is $y_{ij} = \mathrm{DCT}_{ij}^{-1}(\mathbf{c} \cdot \mathbf{q}) \triangleq \sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl} c_{kl}$, $y_{ij} \in \mathbb{R}$. To obtain the final decompressed image, y_{ij} are rounded to integers and clipped to a finite dynamic range $[0, 255]$.

For color images, the RGB representation is typically changed to YC_rC_b (luminance, and two chrominance signals), the luminance Y is processed as above, while the chrominance signals are optionally subsampled, then transformed using DCT, and finally quantized with chrominance quantization matrices, also stored in the header of the JPEG file. For more detailed description of the JPEG format, the reader is referred to [108].

6.3 Analysis

The key idea behind the attack and the first item studied in this section is the statistical distribution of the rounding errors in the spatial domain when decompressing a cover JPEG image. Then, a steganalysis method is developed by testing for this known distribution.

6.3.1 Cover images

We express the decompressed block of non-rounded pixels y_{ij} in terms of the original uncompressed block x_{ij} and the rounding errors in the DCT domain, $u_{kl} \triangleq d_{kl}/q_{kl} - c_{kl}$:

$$\begin{aligned} y_{ij} &= \mathrm{DCT}_{ij}^{-1}(\mathbf{c} \cdot \mathbf{q}) \\ &= \mathrm{DCT}_{ij}^{-1}(\mathbf{d}) - \mathrm{DCT}_{ij}^{-1}(\mathbf{u} \cdot \mathbf{q}) \\ &= x_{ij} - \mathrm{DCT}_{ij}^{-1}(\mathbf{u} \cdot \mathbf{q}) \end{aligned} \tag{6.3.1}$$

where

$$\mathrm{DCT}_{ij}^{-1}(\mathbf{u} \cdot \mathbf{q}) = \sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl} u_{kl}. \tag{6.3.2}$$

Assumption A1: For further analysis, we make the following assumption regarding the rounding errors of cover images in the DCT domain:

$$u_{kl} \sim \mathcal{U}[-1/2, 1/2) \tag{6.3.3}$$

$$u_{kl} \text{ mutually independent.} \tag{6.3.4}$$

for all k, l.

From the independence of u_{kl} and the fact that $E[u_{kl}] = 0$, $Var[u_{kl}] = 1/12$ for all k, l, Lindeberg's extension of the Central Limit Theorem (CLT) implies that y_{ij} approximately follows the Gaussian distribution

$$y_{ij} \sim \mathcal{N}(x_{ij}, s_{ij}), \tag{6.3.5}$$

with variance

$$s_{ij} = \frac{1}{12} \sum_{k,l=0}^{7} (f_{kl}^{ij})^2 q_{kl}^2. \tag{6.3.6}$$

Because x_{ij} is an integer, from Eq. (6.3.1) the rounding error in the spatial domain, $e_{ij} = y_{ij} - [y_{ij}]$, follows the Gaussian distribution $\mathcal{N}(0, s_{ij})$ "folded" to $[-1/2, 1/2)$, which we denoted in Section 8.2 as $e_{ij} \sim \nu(x; s_{ij})$.

6.3.2 Stego images

We model the impact of JPEG-domain steganography as adding a random variable η_{kl} with range $\{-1, 0, 1\}$ to the quantized DCT coefficients $c_{kl} \to c_{kl} + \eta_{kl}$. Assuming $\Pr\{1\} = \Pr\{-1\} = \beta_{kl}$, values β_{kl} are the so-called change rates (the selection channel) determined by the stego scheme. Thus, the decompressed non-rounded stego image z_{ij} is

$$\begin{aligned} z_{ij} &= \mathrm{DCT}_{ij}^{-1}((\mathbf{c} + \boldsymbol{\eta}) \cdot \mathbf{q}) \\ &= \mathrm{DCT}_{ij}^{-1}(\mathbf{d}) + \mathrm{DCT}_{ij}^{-1}((\boldsymbol{\eta} - \mathbf{u}) \cdot \mathbf{q}) \\ &= x_{ij} - \mathrm{DCT}_{ij}^{-1}(\mathbf{u} \cdot \mathbf{q}) + \mathrm{DCT}_{ij}^{-1}(\boldsymbol{\eta} \cdot \mathbf{q}). \end{aligned} \tag{6.3.7}$$

Assumption A2. The embedding changes η_{kl} are independent of the rounding errors u_{kl} and also mutually independent. This is a reasonable assumption for steganography that does not use the rounding errors as side-information for embedding.

Employing the CLT again,

$$z_{ij} \sim \mathcal{N}(x_{ij}, s_{ij} + r_{ij}), \tag{6.3.8}$$

$$r_{ij} = \sum_{k,l=0}^{7} (f_{kl}^{ij})^2 q_{kl}^2 Var[\eta_{kl}]. \tag{6.3.9}$$

Thus, the rounding error of the decompressed stego image, $e_{ij} = z_{ij} - [z_{ij}] \sim \nu(x; s'_{ij})$ with a larger variance $s'_{ij} = s_{ij} + r_{ij}$.

For example, for J-UNIWARD [74] and UED [66, 67], $Var[\eta_{kl}] = \beta_{kl}^{+} + \beta_{kl}^{-}$, where $\beta_{kl}^{+/-}$ are the change rates for changes ± 1 from the embedding simulator or the Syndrome-Trellis Code (STC) [41].

For nsF5 [57] with change rate $\beta_{kl} = \beta$ applied to non-zero AC DCTs, $Var[\eta_{kl}] = \beta$ whenever $(k, l) \neq (0, 0)$ and $c_{kl} \neq 0$.

6.3.3 Hypothesis test

The analysis carried out in the previous two subsections allows us to formulate a statistical hypothesis test for detection of steganography using rounding errors. Given a JPEG block decompressed to the spatial domain but not rounded, z_{ij}, the steganalyst is facing the following hypothesis test for all $0 \leq i, j \leq 7$:

$$\mathrm{H}_0 : e_{ij} \sim \nu(x; s_{ij}) \tag{6.3.10}$$

$$\mathrm{H}_1 : e_{ij} \sim \nu(x; s_{ij} + r_{ij}),\ r_{ij} > 0. \tag{6.3.11}$$

This test is composite if r_{ij} is not known, which would be the case when detecting potentially multiple steganographic methods and / or unknown payload size. On the other hand, for detecting a known steganography and a known payload size, the selection channel is approximately available – the change rates β_{kl} can be computed from the analyzed stego image – which means that r_{ij} can also be approximately computed. Finally, notice that the pair (i, j) is called the "JPEG phase" [73, 72, 119, 19].

Assuming r_{ij} is known and $r_{ij} \ll s_{ij}$, the leading term in the log-likelihood ratio test for the simple hypothesis test (6.3.10) for a single pixel i, j with rounding error e_{ij} is an energy detector:

$$L(e_{ij}) = \log \frac{\nu(e_{ij}; s_{ij} + r_{ij})}{\nu(e_{ij}; s_{ij})} \doteq -\frac{r_{ij}}{2s_{ij}} + \frac{r_{ij}}{2s_{ij}^2} e_{ij}^2. \tag{6.3.12}$$

Next, we focus on JPEG quality 100 and then consider generalizations to lower quality factors.

6.3.4 Quality factor 100

For quality factor 100, $q_{kl} = 1$ for all k, l. Since $\sum_{k,l=0}^{7} (f_{kl}^{(i,j)})^2 = 1$ due to the orthonormality of the DCT, $\mathrm{DCT}_{ij}^{-1}(\mathbf{u} \cdot \mathbf{q}) \sim \mathcal{N}(0, 1/12)$ for all pixels i, j, and $y_{ij} - [y_{ij}] \sim \nu(x; 1/12)$, $x \in [-1/2, 1/2)$:

$$\nu(x; 1/12) = \sqrt{\frac{6}{\pi}} \sum_{n \in \mathbb{Z}} \exp\left(-6(x+n)^2\right) \tag{6.3.13}$$

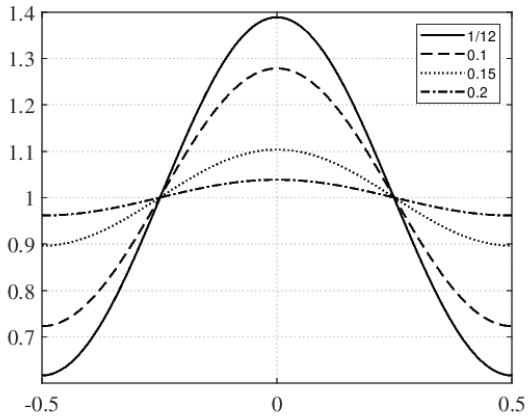

Figure 6.3.1: Distribution $\nu(x;s)$ for $s = 1/12, 0.1, 0.15, 0.2$. Note how rapidly $\nu(x;s)$ converges to a uniform distribution with increased s (also c.f. Tables 6.1–6.2).

shown in Figure 6.3.1. The infinite sum is well approximated[2] with only three terms, $n \in \{-1, 0, 1\}$:

$$\begin{aligned}\nu(x;1/12) &\doteq \sqrt{\frac{6}{\pi}}e^{-6x^2}\left(1+e^{-6}(e^{12x}+e^{-12x})\right)\\ &= \sqrt{\frac{6}{\pi}}e^{-6x^2}\left(1+2e^{-6}\cosh(12x)\right).\end{aligned} \tag{6.3.14}$$

To demonstrate the performance of the energy detector (6.3.12) for this quality factor, we report the results on BOSSbase 1.01 [4] consisting of 10,000 grayscale 512×512 images compressed with Matlab's `imwrite` and embedded with the nsF5 algorithm [57] at 0.2 bpnzac (bits per non-zero AC DCT coefficient). Figure 6.3.2 left shows the distribution of the standard deviation of rounding errors in the spatial domain across all 10,000 cover and stego images while the right graph shows the ROC curve based on this test statistic. The thin right tail of the test statistic across covers gives the detector power close to 0.9 at zero false alarm. The thick left tail is due to the failure of natural images to satisfy Assumptions A1–A2. While we observed $\nu(x;1/12)$ to be a great fit to the distribution of rounding errors for most cover images, our modeling assumptions break, e.g., for images with saturated regions. Additionally, for some images the rounding error for some DCT modes fails to be uniform.

6.3.5 General quality factors

First, notice that for quality less than 100, the distribution of the inverse DCT of the rounding errors u_{kl} depends on the location i, j of the pixel in the block, its JPEG phase. Since the coefficients (6.2.7) in the DCT satisfy

$$|f_{kl}^{ij}| = |f_{kl}^{7-i,j}| = |f_{kl}^{i,7-j}| = |f_{kl}^{7-i,7-j}|, \tag{6.3.15}$$

the variance s_{ij} (6.3.6) inherits the same symmetries

$$|s_{ij}| = |s_{7-i,j}| = |s_{i,7-j}| = |s_{7-i,7-j}|. \tag{6.3.16}$$

Thus, technically the test needs to be applied separately across 16 four-tuples of JPEG phases. However, with decreasing quality factor, the quantization steps q_{kl} increase and thus the variance s_{ij} increases as well. With increased s_{ij}, the folded Gaussian distribution $\nu(x;s_{ij})$ rapidly approaches

[2] With an error less than $\delta = 2.74 \times 10^{-6}$ on the domain of ν (from (6.2.5)).

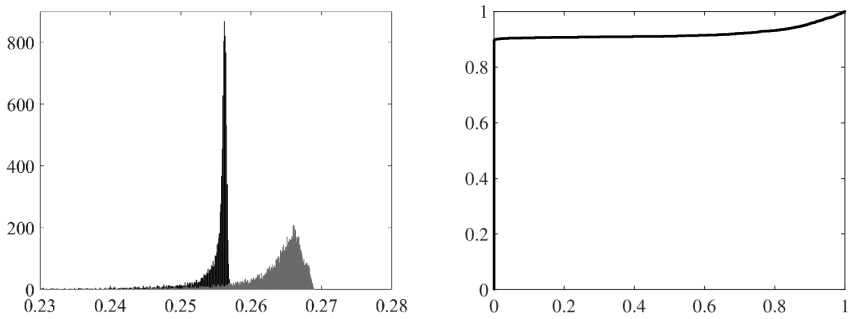

Figure 6.3.2: Left: Distribution of standard deviation of rounding errors for cover QF 100 images (black) and stego images (gray) embedded at 0.2 bpnzac with nsF5. Right: The corresponding ROC curve. Dataset: 10,000 BOSSbase grayscale 512×512 images.

Table 6.1: Minimum and maximum of $\nu(x, s)$ on $[-1/2, 1/2)$ as a function of variance s.

s	$\min \nu$	$\max \nu$
0.083	0.617	1.139
0.10	0.723	1.279
0.15	0.896	1.104
0.20	0.961	1.0386
0.24	0.9825	1.0175
0.30	0.9946	1.0054

$\mathcal{U}[-1/2, 1/2)$ (see Figure 6.3.1), which is why this steganalysis method ceases to be effective. Table 6.1 shows the minimum and maximum values of $\nu(x; s)$ on its domain $[-1/2, 1/2)$ computed for a range of variances s. Additionally, in Table 6.2 we display the minimum and maximum variance s_{ij} across JPEG phases (i, j) for decreasing quality factors.

The attack using rounding errors should still be generally effective for quality 99 because $\nu(x; s)$ is still rather far from a uniform distribution (c.f., Table 6.1–6.2 and Figure 6.3.1). For quality less than 99, however, $\nu(x; s)$ is so close to a uniform distribution that the attack does not work. For quality 98, this attack might still work but only when considering the rounding errors at phases $(i, j) \in \{(0, 0), (0, 7), (7, 0), (7, 7)\}$ for which the variance $s_{ij} \approx 0.24$.[3] This, however, decreases the size of available samples for the test by a factor of 16.

6.4 Machine learning based detectors

Due to the complexity of natural images, Assumptions A1–A2 are satified to a varying degree, which limits the accuracy that can be achieved with detectors derived from idealized models. This

[3]For all other phases, $s_{ij} > 0.37$, which essentially prevents the attack for typical image sizes (see Table 6.1).

Table 6.2: Minimum and maximum variances s_{ij} over JPEG phases i, j for decreasing quality factors.

QF	$\min_{ij} s_{ij}$	$\max_{ij} s_{ij}$
100	0.083	0.083
99	0.105	0.204
98	0.2400	0.822
97	0.492	1.800
96	0.877	3.216

Table 6.3: Detection accuracy of three detectors trained on rounding errors and a conventional SRNet trained on decompressed JPEGs for J-UNIWARD and a range of payloads. BOSSbase + BOWS2 dataset.

	QF 99				QF 100			
bpnzac	e-SRNet	e-GFR	e-Hist	SRNet	e-SRNet	e-GFR	e-Hist	SRNet
0.4	0.9980	0.9840	0.9376	0.8592	0.9998	0.9991	0.9933	0.8829
0.3	0.9960	0.9698	0.9035	0.8054	0.9998	0.9988	0.9865	0.8331
0.2	0.9832	0.9264	0.8376	0.7257	0.9998	0.9967	0.9702	0.7548
0.1	0.9316	0.8284	0.7218	0.6015	0.9998	0.9860	0.9212	0.6488
0.05	0.7989	0.6983	0.6239	0.5437	0.9946	0.9327	0.8486	0.5682

motivated the authors to study machine-learning based detectors trained on the rounding errors in the spatial domain, $e_{ij} = z_{ij} - [z_{ij}]$, where z_{ij} is the decompressed but not rounded (or clipped) JPEG image. Computing the rounding errors can also be viewed as a way to suppress content and form a "noise residual."

This section describes the datasets and detectors that will be used in Section 6.5 containing the results and interpretations of all experiments.

6.4.1 Dataset

Two datasets were used for our experiments. The first is the union of BOSSbase 1.01 and BOWS2, each with 10,000 grayscale images resized to 256×256 pixels with `imresize` in Matlab with default parameters. This dataset is a popular choice for designing detectors with deep learning because small images are more suitable for training deep architectures [133, 8, 134, 135, 130, 140]. The second dataset was prepared from RAW images made available to ALASKA competitors and is detailed in Section 6.6.1.

6.4.2 Detectors

Three types of detectors were implemented: the SRNet [8], a deep convolutional neural network recently proposed for steganalysis in both spatial and JPEG domain, the Gabor Filter Residual features (GFR) [119] with the FLD-ensemble [97] as the classifier, and a feature set consisting of histograms of absolute values of rounding errors split by JPEG phase and symmetrized, also coupled with the ensemble classifier. Since all these detectors were trained on rounding errors e_{ij}, we abbreviate them as e-SRNet, e-GFR, and e-Hist. Next, we describe the details of each classifier and its training.

6.4.2.1 SRNet

For experiments on the union of BOSSbase and BOWS2, all 10,000 BOWS2 images were included in the training set, together with 4,000 randomly selected images from BOSSbase. The validation and testing set, each with 1,000 and 5,000 images were randomly selected from the remaining images from BOSSbase. The training was done for a total 25k iterations with batches of size 64 with an initial LR 2×10^{-3} that was dropped to 2×10^{-4} after 5k iterations.

6.4.2.2 GFR and histograms

The GFR features were extracted from the rounding errors in the spatial domain. This feature set was included as a representative of the class of JPEG phase-aware features, which are among the most powerful rich models for JPEG steganalysis.

Inspired by the analysis from Section 6.3, we developed a third feature representation consisting of quantized histograms of absolute values of rounding errors split by JPEG phase but merged (symmetrized) across phases with the same variance (6.3.16). Formally, denoting the rounding errors in a decompressed 8×8 block of pixels $e_{ij} = z_{ij} - [z_{ij}]$, with $0 \leq i, j < 7$ being the JPEG phase,

$$h_m^{(i,j)} = \left| \left\{ (i', j') \in \mathcal{P}_{ij} \middle| mq \leq |e_{i'j'}| \leq (m+1)q \right\} \right|, \tag{6.4.1}$$

where $q = 1/K$ is a quantization bin width with K a positive integer, $0 \leq m < K/2$ the index of the histogram bin, and $\mathcal{P}_{ij}$ the set of all pixels in a $N_1 \times N_2$ image with phase (i, j): $(i', j') \in \mathcal{P}_{ij}$ if and only if $0 \leq i' < N_1$, $0 \leq j' < N_2$ and $\mathrm{mod}\,(i' - i, 8) = \mathrm{mod}\,(j' - j, 8) = 0$. All 64 K-dimensional histograms (6.4.1) are finally symmetrized to 16 histograms $\tilde{h}_m^{(i,j)}$ based on the symmetry of variances of rounding errors (6.3.16):

$$\tilde{h}_m^{(i,j)} = h_m^{(i,j)} + h_m^{(7-i,j)} + h_m^{(i,7-j)} + h_m^{(7-i,7-j)}, \tag{6.4.2}$$

for $0 \leq i, j < 4$.

The detectors for both the GFR features and the symmetrized phase-split histograms were trained on all images not used for testing of the SRNet as described above, i.e., the training set consisted of 15,000 images for the union of BOSSbase and BOWS2. Finally, we note that $K = 10$ was used for the histograms.

6.5 Experiments

All experiments in this section were executed on the union of BOSSbase and BOWS2 datasets. We first show that the SRNet trained on rounding errors provides better detection than GFR or histograms on rounding errors. Further detection boost is obtained when training the SRNet on two channels – rounding errors and decompressed images, especially for QF 99. We also study the universality of the attack by showing that a detector trained on one embedding scheme can detect other, previously unseen schemes rather well as long as the SRNet is trained only on rounding errors. Finally, we investigate the robustness of this attack w.r.t. different JPEG compressors. Training on the compressor from Python's PIL generalizes overall the best.

6.5.1 Identifying the best detector

First, we studied the performance of the three machine learning detectors trained on rounding errors for quality 99 and 100 for J-UNIWARD and payloads 0.05–0.4 bpnzac. For comparison, in Table 6.3 we also included the results of the conventional SRNet trained on decompressed JPEG images without rounding, which is the established way of training a detector for JPEG images. For quality 100 and payloads 0.2–0.4 bpnzac, the e-SRNet is only slightly better than e-GFR (this is also due to the accuracy being very close to 1). With decreasing payload, however, e-SRNet offers better accuracy than e-GFR by up to 6% for the smallest tested payload. The phase-split histograms (e-Hist) start lagging behind e-SRNet as well as e-GFR increasingly more as the payload size decreases, with the largest loss of 14.6% w.r.t. the e-SRNet for the smallest payload 0.05 bpnzac. Note that the conventional SRNet is markedly less accurate across all payloads with the loss w.r.t. e-SRNet ranging from 11% for the largest payload to 43% for the smallest payload.

For quality 99, the e-SRNet is less accurate than for quality 100 especially for smaller payloads but still detects payload 0.4 bpnzac with 99.80% accuracy. The difference between e-SRNet and e-GFR is much larger than for quality 100. Similar to the quality 100, the phase-split histograms e-Hist and the conventional SRNet are markedly worse.

Table 6.4: Detection accuracy of three different versions of SRNet when training on decompressed images (SRNet), rounding errors (e-SRNet), and both (eY-SRNet). Dataset: BOSSbase + BOWS2.

	QF 99			QF 100		
Payload	SRNet	e-SRNet	eY-SRNet	SRNet	e-SRNet	eY-SRNet
0.4	0.8592	0.9980	0.9994	0.8829	0.9998	0.9995
0.3	0.8054	0.9960	0.9990	0.8331	0.9998	0.9998
0.2	0.7257	0.9832	0.9981	0.7548	0.9998	0.9993
0.1	0.6015	0.9316	0.9780	0.6488	0.9998	0.9984
0.05	0.5437	0.7989	0.9287	0.5682	0.9946	0.9992

For quality 98, all three detectors trained on rounding errors were essentially randomly guessing with the exception of the three largest payloads 0.2–0.4 bpnzac where the e-SRNet achieved accuracy 0.53–0.57, respectively, at which point the conventional SRNet becomes much more accurate. This rapid loss of detection power is to be expected based on the analysis from Section 6.3.

Next, we studied whether the performance of e-SRNet can further be improved by including the decompressed (non-rounded and non-clipped) JPEG image as a second channel (eY-SRNet). Having to train twice as many parameters in the first layer, the eY-SRNet did not converge from scratch for smaller payloads. This was addressed by curriculum learning via payload by first training on the largest payload with batch size 64 for 50k iterations with LR 2×10^{-3}, which was dropped to 2×10^{-4} for 25k more iterations. This detector is then used as a seed for training detectors for smaller payloads with the larger LR for 25k iterations, followed by 25k iterations with the smaller LR.

Table 6.4 shows a clear benefit of using the second channel for QF 99 (eY-SRNet), especially for smaller payloads. For QF 100, the comparison is not as clear because the detection accuracy of both e-SRNet and eY-SRNet is close to 100%.

6.5.2 Universality

Based on the analysis in Section 6.3, we expect the power of the proposed reverse compatibility attack to depend mostly on the payload size and less on the specifics of the steganographic algorithm. In this section, we evaluate the ability of e-SRNet and eY-SRNet to detect steganographic algorithms on which it was not trained on.

Three embedding algorithms were intentionally selected with vastly different embedding operations: nsF5, J-UNIWARD, and Jsteg modified to pseudo-randomly spread non-coded message bits across all DCT coefficients not equal to 0 or 1. First, a detector was trained on a small payload embedded with one of the three stego schemes and then tested on the other two. The payload for each embedding method was empirically selected so that all three embedding schemes exhibit approximately the same detectability, which is not too close to 100% or a random guesser. In particular, the detector for Jsteg was trained on payload 0.01 bpnzac, nsF5 on 0.045 bpnzac, and J-UNIWARD on 0.05 bpnzac. The results are summarized in Figure 6.5.1 showing the missed-detection probability when training on Jsteg (top), nsF5 (middle), and J-UNIWARD (bottom) and testing on stego images embedded with a range of payloads.

For QF 100 (right), the two-channel eY-SRNet performed overall better than e-SRNet. All three detectors generalize to unseen embedding very well with the detector trained on J-UNIWARD being the best. For QF 99 (left), however, e-SRNet generalizes far better than the two channel eY-SRNet, indicating perhaps that it over-specializes on the trained algorithm. Similar to QF 100, the detector trained on J-UNIWARD generalizes the best and also has the smallest false-alarm rate.

Table 6.5: Testing accuracy of e-SRNet trained and tested on JPEGs for all combinations of five JPEG compressors for quality 100, J-UNIWARD 0.05 bpnzac, BOSSbase + BOWS2. The last row shows the performance of eY-SRNet when training on PIL JPEGs.

e-SRNet	Tested on images				
Trained	Matlab	Convert	Int	Float	PIL
Matlab	.9946	.9786	.9953	.9754	.9949
Convert	.8104	.9962	.8103	.9963	.8102
Int	.9964	.9823	.9960	.9790	.9963
Float	.7568	.9970	.7567	.9967	.7567
PIL	**.9959**	**.9889**	**.9966**	**.9879**	**.9959**
eY-SRNet					
PIL	.9974	.9877	.9974	.9874	.9976

6.5.3 Robustness to JPEG compressors

Since there exist many variants of JPEG compressors, which differ mainly in the implementation of the DCT and the internal number representation, the same JPEG image may decompress slightly differently depending on the exact implementation of the DCT, and the same uncompressed image may be compressed to different JPEG files. Such differences may negativelly affect the accuracy of a detector that requires a training set, especially one trained on rounding errors. In this section, we investigate this issue by purposely training on JPEG images obtained with one compressor and testing on images generated by another compressor. We do so for the embedding algorithm J-UNIWARD at quality 100 and payload 0.05 bpnzac.

The following compressors were included in our test: Matlab's imwrite, Python3 library PIL (PIL), ImageMagick's Convert (Convert), Int and Float DCT compressors in libjpeg (version 6b).[4] Fast DCT compression in libjpeg has not been included in our test because it is not recommended for quality factors larger than 97 since the compression is then slower and more lossy than on smaller quality factors.[5]

Table 6.5 shows the complete confusion matrix for quality factor 100 for e-SRNet. While a loss can indeed be observed especially in the case when the detector was built with images generated by 'Float DCT' and 'Convert', the detector trained on images from Python's PIL (boldface in the table) generalized overall very well when evaluated on images from all five compressors. With PIL generalizing the best, we also include the results for the two-channel eY-SRNet trained on images compressed by PIL to verify that adding the decompressed image as a second channel does not negatively affect robustness to different JPEG compressors.

6.6 Experiments on ALASKA

To see how the reverse JPEG compatibility attack performs in more realistic conditions, we include extensive experiments on the ALASKA dataset, which contains color JPEG images of variable size, a diverse cover source with a wide spectrum of processing, four different types of stego algorithms, and variable payload size.

6.6.1 Dataset

We started with 49,928 images acquired in the RAW format provided as part of the steganalysis competition ALASKA. Available from the same web site is the script for converting RAW images

[4] http://libjpeg.sourceforge.net/
[5] Taken from libjpeg documentation https://manpages.ubuntu.com/manpages/artful/man1/cjpeg.1.html.

to JPEGs and for embedding JPEG covers with secret messages. The conversion script develops a RAW image using four different settings and applies varying amounts of sharpening, denoising, resizing, cropping, and micro-contrast enhancement. The final size of the cover image is $N_1 \times N_2$ pixels, where $N_1, N_2 \in \{512, 640, 720, 1024\}$, obtained via "smart" crop that tries to preserve the histogram of local pixel variances (see [58] detailing the smart crop).

The embedding script selects four steganographic methods: J-UNIWARD [74], UED [66], nsF5 [57], and EBS [127] without side information, with priors 0.40, 0.30, 0.15, and 0.15, respectively. The payload size is determined by the processing chain applied by the conversion script when converting the RAW image to JPEG to obtain an approximately constant statistical detectability across various processing chains and JPEG quality factors. For example, the payload is adjusted by considering the image size based on the square root law [89]. All four embedding methods were adjusted to embed in luminance and both chrominance channels as described in [22]. The reader is referred to the above-cited ALASKA web site for more information about both scripts.

6.6.2 Training

Most deep learning architectures proposed for steganalysis [112, 131, 140, 133, 134, 135, 99] cannot be trained on large images because of the memory limitations of current GPUs (11 or 12 GB). For a sufficiently large minibatch size, the images are usually limited to 256×256 or 512×512 pixels. To train a version of the SRNet that can handle images of arbitrary size, such as those from the ALASKA dataset, we adopted a similar approach as in [58] in which first a "tile detector" is trained[6] as a cover-vs-all-stego classifier on 256×256 tiles and then only its Inner Product (IP) classifier layer is retrained on images of arbitrary size. Since the input to the IP layer in SRNet are global means of 512 feature maps outputted by the last convolutional layer, the IP layer is always presented with a 512-dimensional "feature vector" independently of the image size.

The database of RAW images was split into two disjoint parts $\mathcal{T}$ and $\mathcal{I}$, with $\mathcal{T}$ consisting of 39,188 images and $\mathcal{I}$ with 10,740 images. The images from $\mathcal{T}$ were developed with the conversion script modified to output 256×256 smart crops and were used to train the tile detector. The images from $\mathcal{I}$ were processed with the conversion script to arbitrary size and were used for retraining the IP layer on arbitrary sized images. A small portion of $\mathcal{I}$ was also used for validating the tile detector as explained next.

While validating the tile detector on 256×256 tiles was giving us 100% accuracy most of the time, we did observe different performance for different checkpoints after retraining the IP layer on arbitrary size. Thus, we validated the tile detector on arbitrary sized images from $\mathcal{I}$ as this gave more meaningful feedback to select the best checkpoint. This type of validation had to be carried out on batches consisting of one cover-stego pair because in TensorFlow framework, it is not possible to put images of different sizes in one batch.

All images in $\mathcal{T}$ were used for training the tile detector. The breakup of $\mathcal{I}$ into TRN / VAL / TST and VAL for the tile detector was 3,656 / 1,500 / 4,000, and 1,584.

The tile detector training was carried out for a total 30k iterations with mini batch size 64, starting with Learning Rate (LR) 2×10^{-3}, which was dropped to 2×10^{-4} after 10k iterations. The IP layer was retrained for 20k iterations with LR 10^{-3} and batch size 800. The setting for this layer was kept the same as for the IP layer at the end of the tile detector.

6.6.3 Searching for the best detector

Since the images in ALASKA are color, our first test was aimed at investigating whether the chrominance channels help improve detection accuracy of e-SRNet trained on rounding errors. In particular, we tested a three-channel variant of the e-SRNet in which all 64 3×3 filters in the first layer were

[6] The batches were formed with the same priors as in the ALASKA dataset (Section 6.6.1).

replaced with 64 $3 \times 3 \times 3$ filters applied to rounding errors of the luminance and both chrominance signals. As Table 6.6 shows, however, the three-chanel e-SRNet gave essentially the same results as using only the luminance. This is surprising since the conventional SRNet on quality factors other than 99 and 100 greatly benefited from including the chrominance channels [137]. We hypothesize that this is due to two reasons. First, for QFs near 100 in the ALASKA dataset, the chrominance channel carries only one half of the total payload as the luminance and thus affects the distribution of the folded Gaussian to a lesser degree. Second, since the chrominance has a narrower dynamic range than luminance, the rounding error in the DCT domain is not uniform, further violating Assumption A1 (Section 6.3.1) under which the reverse JPEG compatibility attack was derived. All remaining experiments on ALASKA were thus executed with luminance only.

The focus of the next round of exploration was to determine whether the following design choices might perhaps further improve the detector performance:

1. Supplying the non-rounded image as a second channel (eY-SRNet)
2. Training the tile detector as multi-class instead of cover-vs.-all-stego
3. Using four moments of feature maps outputted by the tile detector to better handle images of arbitrary size
4. Using MLP instead of IP layer for the arbitrary size detector.

As observed in the previous section, while the two-channel eY-SRNet performed better than e-SRNet, it was also less robust w.r.t. a stego-source mismatch, i.e., when testing on an unseen embedding algorithm. Since the ALASKA dataset contains stego images from four different embedding schemes, it can be expected that the more robust e-SRNet will give better results than eY-SRNet. This was, indeed, confirmed experimentally as shown in Table 6.7. This table shows the probability of correct detection of three different versions of SRNet achieved on the ALASKA dataset with stego images following their corresponding priors, on covers (this is essentially $1 - P_{\text{FA}}$), and then on each embedding algorithm. Note that the lowest detection accuracy is for nsF5, which is due to the payload scaling applied in ALASKA (nsF5 stego images have the smallest payload).

To address the second item above, we recall the results reported in [11] on steganalysis of diversified stego sources. The authors investigated several methodologies for building a detector for stego source containing images from seven different steganographic schemes in the spatial domain. In particular, training the SRNet as multi-class (but using as binary to distinguish stego images from covers) gave better results than training it as cover-vs.-all-stego. Training the e-SRNet as multi-class, including retraining the IP layer as multi-class, however, did not translate to a gain. In fact, as Table 6.7 shows, the correct detection was lower on ALASKA and on covers with statistically insignificant improvements for J-UNIWARD and UED.

Finally, we only comment on the effect of items 3 and 4 above. Outputting the minimum, maximum, and variance of the feature maps on the tile detector's output, in addition to the global mean, did not lead to any improvement in detection performance. Neither did we observe any gains when replacing the IP layer with a MLP with one hidden layer of double the dimensionality of the output of the tile detector. In summary, the best overall detector for QFs 99 and 100 on the ALASKA dataset was the e-SRNet trained on rounding errors of luminance only with only the global means as output of the tile detector and a simple IP layer retrained on arbitrary sizes. The tile detector as well as the IP (for arbitrary size) were trained as one-vs.-all-stego classifiers on minibatches formed by respecting the priors for the four stego schemes.

6.7 Countermeasures

Fundamentally, the proposed reverse JPEG compatibility attack is possible because the signals entering the DCT in the JPEG compressor are integer-valued. Therefore, a countermeasure against

Table 6.6: Detection accuracy of e-SRNet on ALASKA test set when using only the rounding errors from luminance and a three-channel e-SRNet when using the rounding errors from all three channels.

QF	99	100
Luminance	0.9400	0.9900
Color	0.9375	0.9893

Table 6.7: Probability of correct detection of e-SRNet, eY-SRNet, and multi-class e-SRNet (all on luminance only) on ALASKA, covers ($1 - P_{\text{FA}}$), and each embedding algorithm. Results obtained on $5 \times 4,000$ images from the test set.

	e-SRNet		eY-SRNet		Multi-class e-SRNet	
	QF 99	QF 100	QF99	QF100	QF 99	QF 100
ALASKA	0.9400	0.9900	0.9296	0.9450	0.9098	0.9794
Cover	0.9960	0.9985	0.9960	0.9915	0.9810	0.9993
EBS	0.9550	0.9873	0.9563	0.9788	0.9423	0.9810
JUNI	0.9945	0.9888	0.9690	0.9860	1.0000	0.9985
nsF5	0.2880	0.9508	0.2598	0.3865	0.0395	0.7945
UED	0.9825	0.9875	0.9635	0.9820	0.9910	0.9885

this attack would be to not round the luminance (and chrominances for color) before applying the DCT.

To test this hypothesis, uncompressed (16-bit TIFF) color 256×256 smart crops obtained using the developing script from ALASKA were converted to monochrome images using the relationship $Y = 0.299 \times R + 0.589 \times G + 0.114 \times B$, where R, G, B stand for red, green, and blue channel, respectively, and then scaled to the 8-bit range $[0, 255]$ without rounding. Each image was then processed using block DCT (implemented with `dct2` in Matlab). The resulting DCT coefficients were then quantized with quality 100, rounded, and finally written to a JPEG file using the JPEG Toolbox. These cover JPEGs were then embedded with J-UNIWARD at 0.2 bpnzac. With the same breakup of ALASKA into training, validation, and testing, the e-SRNet achieved the testing accuracy of 55.05, which confirms the effectiveness of this countermeasure.

This countermeasure, however, has a flaw since, to the best knowledge of the authors, all JPEG compressors round the luminance before applying the DCT. Thus, images compressed from non-rounded luminance are rare and should be suspicious by themselves. In other words, the proposed countermeasure only works within an artificially crafted cover source. In fact, since rounding errors of integer-valued compressed images follow the folded Gaussian distribution (see Figure 6.3.1) and the rounding errors of non-integer compressed images do not, both sources can be reliably distinguished: for quality 100, the SRNet tile detector trained on rounding errors of only luminance achieved 100% accuracy. Training was done with mini batch size 64 for 30k iterations, with initial LR 10^{-3} dropped to 10^{-4} after 10k iterations. The best checkpoint was selected after 4k iterations.

A more viable alternative is to break the independence of the rounding error u_{kl} and the embedding change η_{kl} (Assumption A2 in Section 6.3.2) and ensure that the variance of $u_{kl}+\eta_{kl}$ stays as close to $1/12$ as possible. This is exactly what the so-called side-informed embedding schemes [31, 67] achieve heuristically by modulating the costs of changing each DCT coefficient by $1 - 2|u_{kl}|$. Therefore, as the next step, we switched to the BOSSbase + BOWS2 dataset and tested the security of SI-UNIWARD [74] on a range of payloads for both quality factors 99 and 100 when steganalyzed with e-SRNet, eY-SRNet, and a conventional SRNet. In this case, the two-channel eY-SRNet gave overall best performance for QF 99 and QF 100 (see Table 6.8 for the complete results). Comparing the high detectability of J-UNIWARD (Table 6.4), we conclude that SI-UNIWARD is an effective counter measure for the reverse JPEG compatibility attack as long as the payload size is kept below 0.05 bpnzac.

Table 6.8: Accuracy of the conventional SRNet trained on decompressed images (SRNet), e-SRNet on rounding errors, and a two-channel eY-SRNet trained on decompressed images and rounding errors across different payloads of SI-UNIWARD. Dataset: BOSSbase + BOWS2.

	99			100		
bpnzac	SRNet	e-SRNet	eY-SRNet	SRNet	e-SRNet	eY-SRNet
0.4	0.6859	0.9517	0.9960	0.6960	0.9954	0.9941
0.3	0.6106	0.9391	0.9824	0.6186	0.9925	0.9923
0.2	0.5457	0.8354	0.9208	0.5474	0.9758	0.9915
0.1	0.5030	0.6278	0.6862	0.5474	0.8514	0.8978
0.05	0.5000	0.5110	0.5344	0.5291	0.6107	0.6800

6.8 Conclusions

A new compatibility steganalysis attack is proposed, which is applicable to both color and grayscale JPEG images saved with quality 99 and 100. It is based on the observation that, when decompressing a JPEG image, the rounding errors in the spatial domain exhibit a Gaussian distribution with variance 1/12 folded to $[-1/2, 1/2)$. Steganographic embedding changes made to quantized DCT coefficients increase the variance of the Gaussian distribution, allowing thus an extremely accurate detection. The attack is fundamentally possible due to the fact that the DCT is applied to integers.

While the basic principle of the attack is explained and introduced under simplifying modeling assumptions using statistical hypothesis testing, the best detectors in practice are obtained with classifiers trained on rounding errors. Three types of classifiers were investigated – Gabor Filter Residuals, phase-split histograms of rounding errors, and a deep residual network called the SRNet, which consistently provided the best results in our experiments.

The attack has been tested on five different embedding schemes, grayscale and color images, and diverse stego sources (the ALASKA dataset). It appears to be universal in the sense that a detector trained on one embedding algorithm generalizes to unseen embedding methods. The attack is also robust to various JPEG compressors. Moreover, it has been shown that steganalysis targeted to a specific embedding algorithm can be improved, especially for quality factor 99, by providing rounding errors together with decompressed image as input to the network detector.

To circumvent the attack, one needs to avoid applying the DCT to integer-valued images, which, however, none of the JPEG compressors known to the authors do. The second possibility to reduce the detectability is to use side-informed embedding schemes that minimize the combined distortion due to quantization and embedding. They, indeed, are less detectable than non-side-informed schemes. Our experiments showed that SI-UNIWARD on payload of 0.05 bpnzac essentially eluded detection. Thus, besides drastically reducing the payload, it currently appears that quality 100 and 99 JPEGs should be avoided for steganography by the same token as decompressed JPEGs should not be used for spatial-domain embedding.

All code used to produce the results in this paper, including the network configuration files will be made available from `http://dde.binghamton.edu/download/` upon acceptance of this paper.

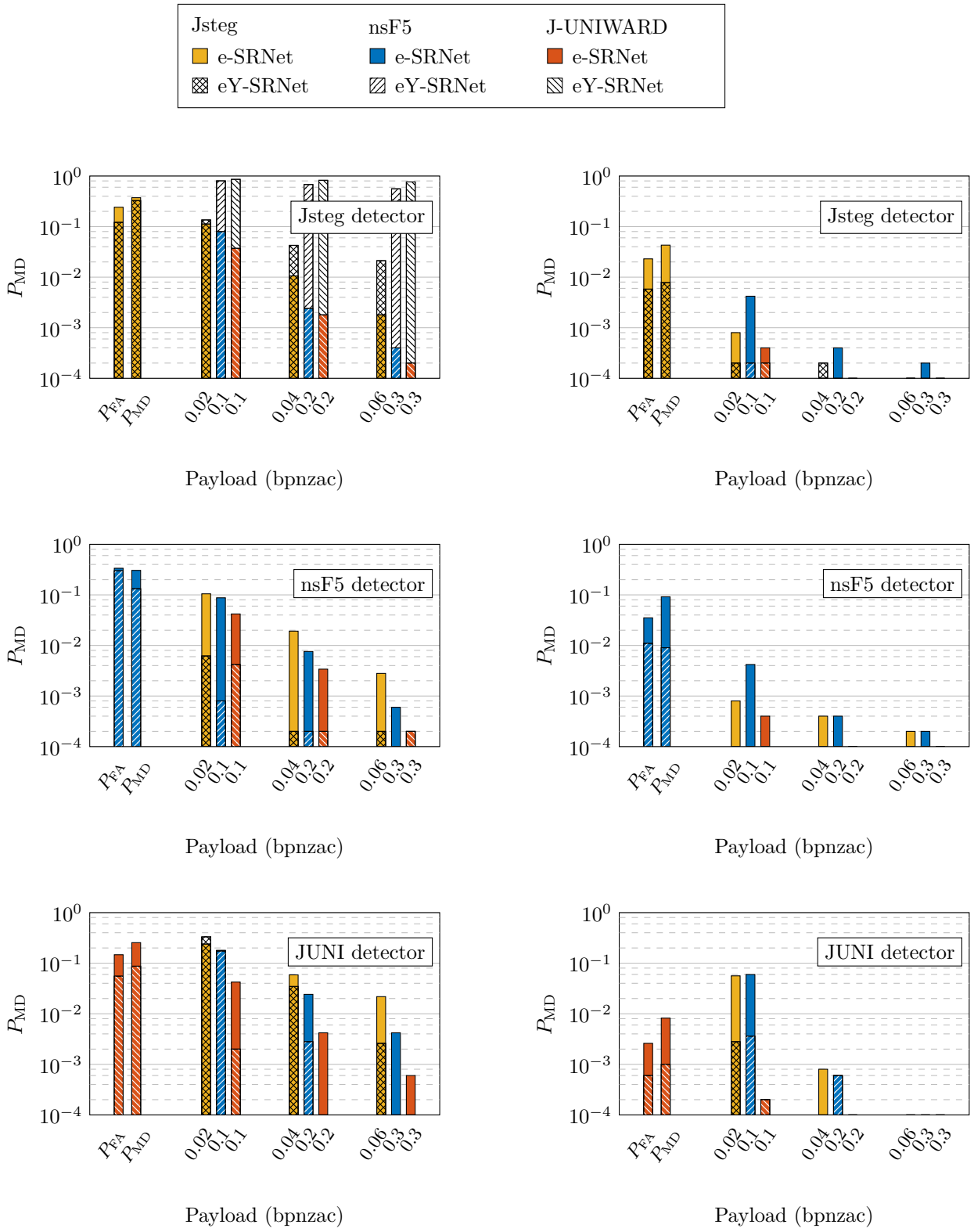

Figure 6.5.1: Probability of missed detection P_{MD} (in logarithmic scale) on stego images embedded with three different stego schemes and payloads when training e-SRNet (color) and eY-SRNet (patterns) for Jsteg (top), nsF5 (middle), and J-UNIWARD (bottom) on payloads 0.01, 0.045, and 0.05 bpnzac, respectively. The first two columns denoted by P_{FA} and P_{MD} correspond to the false-alarm and missed-detection rates of each detector. The value 10^{-4} is used to represent $P_{\text{MD}} = 0$ as this value was never achieved in terms of missed detection. Testing payloads were chosen to be roughly 2, 4 and 6 times of the payload used in training. Left: QF 99, right: QF 100. Dataset: BOSSbase + BOWS2.

Chapter 7

Extending the Reverse JPEG Compatibility Attack to Double Compressed Images

The reverse JPEG compatibility attack has recently been introduced as a very accurate and universal steganalysis algorithm for JPEG images with quality 99 or 100. The limitation to these two largest qualities appears fundamental as the prior work on this topic suggests. In this paper, we provide mathematical analysis and demonstrate experimentally that this attack can be extended to double compressed images when the first compression quality is 93 or larger and the second quality equal or larger than the first quality. Comparisons with state-of-the-art deep convolutional neural networks as well as detectors built in the JPEG domain show the merit of this work.

7.1 Introduction

Recently, a qualitatively new type of attack on JPEG steganography has been introduced, the Reverse JPEG Compatibility Attack (RJCA) [13], which forms the detection statistic from the rounding errors of a decompressed JPEG image. Unlike other detectors, the RJCA is universal (can detect any steganography) and can reliably detect even very short messages. It can thus provide a very high certainty about usage of steganography even from a single intercepted image. The disadvantage of this attack is its limited applicability to only JPEG quality 99 or 100. As the original paper on this topic shows, this limitation is quite fundamental and cannot be overcome due to the nature of the JPEG compression itself.

The main reason why the RJCA works is because, during compression, the discrete cosine transform (DCT) is applied to an integer-valued signal. This allows modeling the rounding errors after decompression as a zero-mean Gaussian with variance $\approx 1/12$ folded into the interval $[-0.5, 0.5)$. The embedding increases the variance of the Gaussian, which begins to fold into a uniform distribution. The attack can be realized by training a classifier on rounding errors [13] or using a simplified likelihood ratio test when the selection channel is known [21]. The attack does not work for lower qualities because the variance of the folded Gaussian increases rapidly with increasing quantization steps, making the distribution of the rounding errors essentially uniform even for cover images.

The main novel idea presented in this paper is the realization that the above-mentioned limitation relates to *single compressed images*, and does not necessarily apply to images that were compressed more than once. We show using mathematical analysis as well as experimentally that the RJCA can be extended to images doubly compressed with qualities $93 \leq Q_1 \leq Q_2$, broadening thus

the applicability of this attack in practice. In particular, the attack is extremely accurate when $Q_1 = Q_2$, when the detectors that do not utilize rounding errors perform poorly. Images doubly compressed with the same quality factor naturally arise due to minor retouching, such as removing wrinkles or sensor dust, and adding a visible watermark when the editing tool is set to preserve the compression parameters. Moreover, double compressed JPEG covers can be introduced either by a conscious action of the sender or inadvertently due to the processing pipeline that precedes the actual embedding.

In the next section, we introduce the notation and preliminary concepts. Section 7.3 analyzes the distribution of rounding errors in the spatial domain after double compression for both cover and stego images. We analytically show that the rounding errors after decompression can again be modeled as a folded Gaussian distribution if either the quality settings during both compression steps are the same or if $Q_2 = 99$ or 100 and $Q_1 \geq 93$. The theoretical insight is put to test in Section 7.4, where we report the detection accuracy of the RJCA for J-UNIWARD and benchmark it against SRNet [8] and JRM [96]. The paper is concluded in Section 7.5.

7.2 Preliminaries and notation

Boldface symbols are reserved for matrices and vectors with elementwise multiplication and division denoted $\odot$ and $\oslash$. The uniform distribution on the interval $[a, b]$ will be denoted $\mathcal{U}[a, b]$ while $\mathcal{N}(\mu, \sigma^2)$ is used for the Gaussian distribution with mean μ and variance σ^2. Rounding x to an integer is denoted $[x]$. The set of all integers will be denoted $\mathbb{Z}$. For $X \sim \mathcal{N}(\mu, \sigma^2)$ with $\mu \in \mathbb{Z}$, the rounding error $X - [X] \sim \mathcal{N}_F$, the Gaussian distribution folded on $-1/2 \leq x < 1/2$, with pdf

$$\nu(x; \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{n \in \mathbb{Z}} \exp\left(-\frac{(x+n)^2}{2\sigma^2}\right). \tag{7.2.1}$$

For better readability, we strictly use i, j to index pixels and k, l DCT coefficients. Denoting by x_{ij}, $0 \leq i, j \leq 7$, an 8×8 block of pixels, they are transformed during JPEG compression to DCT coefficients $d_{kl} = \mathrm{DCT}_{kl}(\mathbf{x}) \triangleq \sum_{i,j=0}^{7} f_{kl}^{ij} x_{ij}$, $0 \leq k, l \leq 7$, and then quantized $c_{kl} = [d_{kl}/q_{kl}]$, $c_{kl} \in \{-1024, \dots, 1023\}$, where q_{kl} are quantization steps in a luminance quantization matrix, and $f_{kl}^{ij} = w_k w_l / 4 \cos \pi k(2i+1)/16 \cos \pi l(2j+1)/16$, $w_0 = 1/\sqrt{2}$, $w_k = 1$, $0 < k \leq 7$, are the discrete cosines.

During decompression, the above steps are reversed. For a block of quantized DCTs c_{kl}, the corresponding block of non-rounded pixels after decompression is $y_{ij} = \mathrm{DCT}_{ij}^{-1}(\mathbf{c} \odot \mathbf{q}) \triangleq \sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl} c_{kl}$, $y_{ij} \in \mathbb{R}$. To obtain the final decompressed image, y_{ij} are rounded to integers and clipped to $[0, 255]$.

For color images, the *RGB* representation is typically changed to YC_bC_r (luminance, and two chrominance signals), the luminance Y is processed as above, while the chrominance signals are optionally subsampled, then transformed using DCT, and finally quantized with chrominance quantization matrices [108].

7.3 Rounding errors and double compression

In this section, we derive a model for the statistical distribution of the rounding errors in the spatial domain when decompressing a doubly compressed cover image and its stego version. The quantization matrices and quality factors used for the first and second compression will be denoted as $\mathbf{q}^{(1)}$, $\mathbf{q}^{(2)}$ and Q_1, Q_2, respectively.

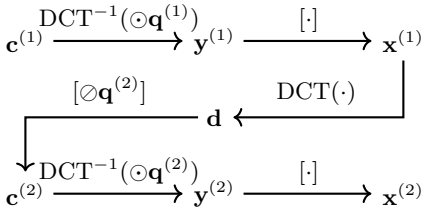

Figure 7.3.1: Double compression pipeline.

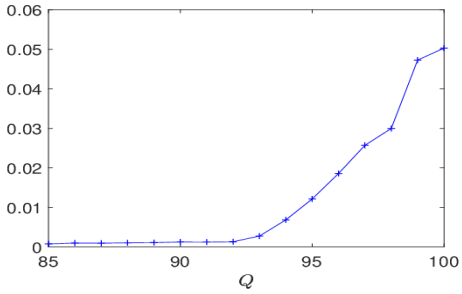

Figure 7.3.2: Relative number of different quantized DCTs when recompressing an image with quality Q with the same quality. Results averaged over 1000 images from BOSSbase 1.01.

7.3.1 Cover images

Starting with a single-compressed JPEG file represented with quantized DCT coefficients $\mathbf{c}^{(1)}$, we consider the pipeline shown in Figure 8.2.1, which consists of decompressing $\mathbf{c}^{(1)}$ to $\mathbf{y}^{(1)}$, rounding to integers $\mathbf{x}^{(1)}$, compressing the second time with quantization matrix $\mathbf{q}^{(2)}$ to obtain DCT coefficients before quantization $\mathbf{d}$ and after quantization $\mathbf{c}^{(2)}$, decompressing to non-rounded pixels $\mathbf{y}^{(2)}$ and rounding to $\mathbf{x}^{(2)}$. Assuming the rounding errors in the spatial domain $u_{ij}^{(1)} = y_{ij}^{(1)} - x_{ij}^{(1)} \sim \mathcal{U}[-1/2, 1/2)$, we have $\mathbb{E}[u_{ij}^{(1)}] = 0$, $Var[u_{ij}^{(1)}] = 1/12$. Since

$$y_{ij}^{(1)} = \mathrm{DCT}^{-1}(\mathbf{c}^{(1)} \odot \mathbf{q}^{(1)}) = \sum_{k,l=0}^{7} f_{kl}^{ij} c_{kl}^{(1)} q_{kl}^{(1)} \tag{7.3.1}$$

$$x_{ij}^{(1)} = y_{ij}^{(1)} - u_{ij}^{(1)} = \sum_{k,l=0}^{7} f_{kl}^{ij} c_{kl}^{(1)} q_{kl}^{(1)} - u_{ij}^{(1)}, \tag{7.3.2}$$

we can write

$$\begin{aligned} d_{kl} &= \mathrm{DCT}_{kl}(\mathbf{x}^{(1)}) \\ &= \mathrm{DCT}_{kl}(\mathbf{y}^{(1)} - \mathbf{u}^{(1)}) \\ &= c_{kl}^{(1)} \cdot q_{kl}^{(1)} - \sum_{i,j=0}^{7} f_{kl}^{ij} u_{ij}^{(1)}. \end{aligned} \tag{7.3.3}$$

Assuming that $u_{ij}^{(1)}$ are mutually independent, from the CLT and orthonormality of the DCT :

$$d_{kl} \sim \mathcal{N}\left(c_{kl}^{(1)} q_{kl}^{(1)}, \frac{1}{12}\right). \tag{7.3.4}$$

Denoting the rounding error in the DCT domain during the second compression as $e_{kl} = d_{kl}/q_{kl}^{(2)} -$

$c_{kl}^{(2)} = d_{kl}/q_{kl}^{(2)} - [d_{kl}/q_{kl}^{(2)}]$, from (7.3.4), e_{kl} follows a folded Gaussian distribution on $[-1/2, 1/2)$

$$e_{kl} \sim \mathcal{N}_F \left(c_{kl}^{(1)} \frac{q_{kl}^{(1)}}{q_{kl}^{(2)}}, \frac{1}{12(q_{kl}^{(2)})^2} \right) \tag{7.3.5}$$

with expectation

$$\mathbb{E}[e_{kl}] = c_{kl}^{(1)} \frac{q_{kl}^{(1)}}{q_{kl}^{(2)}} - \left[c_{kl}^{(1)} \frac{q_{kl}^{(1)}}{q_{kl}^{(2)}} \right]. \tag{7.3.6}$$

Continuing our analysis,

$$\begin{aligned} y_{ij}^{(2)} &= \mathrm{DCT}_{ij}^{-1}(\mathbf{c}^{(2)} \odot \mathbf{q}^{(2)}) \\ &= \mathrm{DCT}_{ij}^{-1}(\mathbf{d} - \mathbf{e} \odot \mathbf{q}^{(2)}) \\ &= \mathrm{DCT}_{ij}^{-1}\left(\mathrm{DCT}(\mathbf{y}^{(1)} - \mathbf{u}) - \mathbf{e} \odot \mathbf{q}^{(2)}\right) \\ &= y_{ij}^{(1)} - u_{ij}^{(1)} - \mathrm{DCT}_{ij}^{-1}(\mathbf{e} \odot \mathbf{q}^{(2)}) \\ &= x_{ij}^{(1)} - \eta_{ij}, \end{aligned} \tag{7.3.7}$$

where $\eta_{ij} = \sum_{k,l=0}^{7} f_{kl}^{ij} e_{kl} q_{kl}^{(2)}$. Assuming the independence of the rounding errors e_{kl}, the CLT implies

$$\eta_{ij} \sim \mathcal{N}\Big(\sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl}^{(2)} \mathbb{E}[e_{kl}], \sum_{k,l=0}^{7} (f_{kl}^{ij})^2 (q_{kl}^{(2)})^2 Var[e_{kl}] \Big). \tag{7.3.8}$$

Thus, $y_{ij}^{(2)}$ follows a Gaussian distribution with mean

$$\mathbb{E}[y_{ij}^{(2)}] = x_{ij}^{(1)} - \sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl}^{(2)} \mathbb{E}[e_{kl}]. \tag{7.3.9}$$

Note that for $q_{kl}^{(2)} > 1$, the variance of the folded Gaussian distribution (8.4.5) is approximately the same as the variance of the Gaussian that e_{kl} follows, $Var[e_{kl}] \approx 1/(12(q_{kl}^{(2)})^2)$, and thus $Var[y_{ij}^{(2)}] \approx 1/12$.

With this approximation, the rounding error after the second decompression $u_{ij}^{(2)} = y_{ij}^{(2)} - x_{ij}^{(2)}$ follows a Gaussian distribution, which is folded into $[-1/2, 1/2)$, with mean and variance

$$\mathbb{E}[u_{ij}^{(2)}] = -\sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl}^{(2)} \mathbb{E}[e_{kl}] + \left[\sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl}^{(2)} \mathbb{E}[e_{kl}] \right] \tag{7.3.10}$$

$$Var[u_{ij}^{(2)}] = 1/12. \tag{7.3.11}$$

For the RJCA to work, the distribution of the rounding error cannot be uniform as in this case, the embedding would not change it. In particular, if the expectations (7.3.10) are not zero and vary across pixels ij, the resulting mixture becomes practically uniform. On the other hand, when $\mathbb{E}[e_{kl}] = 0$ for most DCT modes kl and blocks, $\mathbb{E}[u_{ij}^{(2)}] \approx 0$ and the RJCA works again. Note that from (7.3.6) $\mathbb{E}[e_{kl}] = 0$ when $q_{kl}^{(2)}$ divides $c_{kl}^{(1)} q_{kl}^{(1)}$. Since we need this to be satisfied for the majority of the blocks and irrespectively of the content, we arrive at our first condition:

Q_1	detector	Q_2 = 93	94	95	96	97	98	99	100
93	e-SRNet	0.0438	0.3678	0.4104	0.3545	0.2845	0.0317	0.0002	0.0002
	eOH-SRNet	0.0485	0.0059	0.0019	0.0024	0.0035	0.0051	0.0001	0.0001
	JRM	0.4360	0.0029	0.0028	0.0016	0.0010	0.0031	0.0064	0.0053
94	e-SRNet		0.0028	0.3356	0.4205	0.1725	0.0994	0.0001	0.0000
	eOH-SRNet		0.0027	0.0076	0.0030	0.0033	0.0060	0.0002	0.0001
	JRM		0.4304	0.0023	0.0022	0.0019	0.0022	0.0050	0.0068
95	e-SRNet			0.0009	0.3449	0.2870	0.0463	0	0.0001
	eOH-SRNet			0.0008	0.0008	0.0038	0.0038	0.0002	0.0001
	JRM			0.4232	0.0067	0.0024	0.0039	0.0052	0.0067
96	e-SRNet				0.0006	0.3251	0.0412	0.0001	0.0001
	eOH-SRNet				0.0004	0.0118	0.0062	0.0001	0.0002
	JRM				0.4196	0.0079	0.0058	0.0068	0.0086
97	e-SRNet					0.0005	0.2055	0.0001	0.0003
	eOH-SRNet					0.0003	0.0482	0.0002	0.0001
	JRM					0.4159	0.0207	0.0070	0.0061
98	e-SRNet						0.0003	0.0001	0.0001
	eOH-SRNet						0.0001	0.0002	0.0001
	JRM						0.4194	0.0031	0.0041
99	e-SRNet							0	0.0001
	eOH-SRNet							0.0001	0
	JRM							0.4127	0.0026
100	e-SRNet							0.0002	0.0001
	eOH-SRNet							0.0001	0.0001
	JRM							0.4126	0.3965

Table 7.1: Detection error P_{E} with different detectors, J-UNIWARD at 0.4 bpnzac.

[C1] $q_{kl}^{(2)}$ divides $q_{kl}^{(1)}$ for most modes kl.

Note that this means that $Q_1 \leq Q_2$. Unless both qualities are equal, however, the double-compressed image will exhibit strong signs of double-compression with gaps and peaks in the DCT histogram, which will make steganography highly detectable using standard steganalysis features, such as the JRM [96]. Thus, from now on, we mainly focus on cases when $Q_1 = Q_2$ while noting that the RJCA remains extremely accurate when $Q_2 = 99$ or $Q_2 = 100$.

Moreover, notice that when $\mathbf{c}^{(1)} = \mathbf{c}^{(2)}$, the double-compressed image is the same as the single-compressed image, and, as already established in [13], the RJCA for single-compressed images works only for qualities 99 and 100. Thus, the second condition for the RJCA to work in doubly-compressed images with $Q_1 = Q_2$ is

[C2] $\mathbf{c}^{(1)} \neq \mathbf{c}^{(2)}$,

which is mainly fulfilled if there are ones in the quantization table or equivalently $Q_2 \geq 93$. This is confirmed in Figure 8.6.2 showing the average number of DCT coefficients that changed during recompression with the same quality factor across 1000 images selected from BOSSbase 1.01 at random. This result is not sensitive to the specific implementation of the JPEG compressor.

7.3.2 Stego images

Given a JPEG cover image represented by DCT coefficients $\mathbf{c}^{(1)}$, the steganographer embeds the secret message into the image after recompression $\mathbf{c}^{(2)}$. We model the steganography by adding steganographic noise $\xi_{kl} \in \{-1, 0, 1\}$, $\Pr\{\xi_{kl} = 1\} = \beta^+$, $\Pr\{\xi_{kl} = -1\} = \beta^-$ to the cover: $s_{kl} = c_{kl}^{(2)} + \xi_{kl}$. Note that $\mathbb{E}(\xi_{kl}) = \beta_{kl}^+ - \beta_{kl}^-$ and $Var[\xi_{kl}] = \beta_{kl}^+ + \beta_{kl}^-$.

Decompressing the stego image block gives

$$\begin{aligned} z_{ij} &= \mathrm{DCT}_{ij}^{-1}(\mathbf{s} \odot \mathbf{q}^{(2)}) \\ &= \mathrm{DCT}_{ij}^{-1}(\mathbf{c}^{(2)} \odot \mathbf{q}^{(2)} + \xi \odot \mathbf{q}^{(2)}) \\ &= x_{ij}^{(1)} - \eta_{ij} + \zeta_{ij}, \end{aligned} \tag{7.3.12}$$

where $\zeta_{ij} = \sum_{kl} f_{kl}^{ij} \xi_{kl} q_{kl}^{(2)}$

$$\zeta_{ij} \sim \mathcal{N}\Big(\sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl}^{(2)} (\beta_{kl}^{+} - \beta_{kl}^{-}), \sum_{k,l=0}^{7} (f_{kl}^{ij})^2 (q_{kl}^{(2)})^2 (\beta_{kl}^{+} + \beta_{kl}^{-}) \Big). \tag{7.3.13}$$

For steganography without side information $\beta_{kl}^{+} = \beta_{kl}^{-}$, thus

$$z_{ij} \sim \mathcal{N}\Big(x_{ij}^{(1)} - \sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl}^{(2)} \mathbb{E}[e_{kl}], \sum_{k,l=0}^{7} (f_{kl}^{ij})^2 (q_{kl}^{(2)})^2 (\beta_{kl}^{+} + \beta_{kl}^{-} + Var[e_{kl}]) \Big). \tag{7.3.14}$$

Notice that the rounding error of z_{ij} is a folded Gaussian whose variance is increased due to embedding (c.f. Eq. (7.3.8) with Eq. (7.3.14)) and whose mean is now *non-zero*, dependent on the rounding errors in DCT domain. Both contribute to the fact that in stego images, these Gaussians will start folding into a uniform distribution with increased payload (change rates).

7.4 Results

All experiments in this paper are executed on the union of the popular datasets BOSSbase 1.01 and BOWS2, each containing 10,000 grayscale images downsampled to 256×256 using 'imresize' with default parameters in Matlab. The detectors were trained on all BOWS2 images and a randomly selected 4,000 BOSSbase images, with 1,000 BOSSbase images used for validation and 5,000 for testing.

Table 7.1 shows the detection error under equal priors on the testing set for J-UNIWARD at 0.4 bpnzac. The cover JPEG images were doubly compressed with the first quality factor being represented by rows and the second quality factor by columns. We only show the cases when $93 \leq Q_1 \leq Q_2$ and also when $Q_1, Q_2 \in \{99, 100\}$, since these cases satisfy condition [C1]. Three detectors are tested: SRNet [8] trained on the rounding errors after decompressing the JPEG image (e-SRNet), JRM with the ensemble classifier [97], and OneHot network [139] combined with e-SRNet (eOH-SRNet), which is implemented as OneHot-SRNet in the original paper with clipping threshold $T = 5$. The SRNet, however, takes the rounding errors on the input instead of the spatial representation of the image. We want to point out that both network based detectors converge to their optimum extremely quickly, within 20k iterations. Even though e-SRNet fails for some combinations of the compression qualities, such as (96, 97), double compression with such combinations of quality factors leads to peaks and valleys in cover DCT histograms, which allows very accurate detection with JRM and other prior art [29, 132, 142, 139]. Note that these detectors perform rather poorly whenever $Q_1 = Q_2$. The eOH-SRNet provides overall reliable detection.

The condition [C2] dictates that the RJCA will work whenever the (equal) quality factors are at least 93 and that can be confirmed in Table 7.1. Results for lower qualities are not included because RJCA stops working there, in agreement with the analysis from Section 7.3.

7.5 Conclusions

The reverse JPEG compatibility attack is an extremely accurate, universal, and quite simple steganalysis technique that was originally shown to be limited to high quality factors (99 or 100). In

this paper, we extend this attack to cover images that are doubly compressed with quality factors $93 \leq Q_1 \leq Q_2$. By analyzing the distribution of the rounding errors in the spatial domain, we arrived at two conditions that need to be satisfied for the attack to work. The conclusions reached from the theoretical considerations match our experimental results. In combination with the OneHot-SRNet, the detector provides the most reliable detection across all above combinations of quality factors. In particular, the compatibility attack works extremely reliably also when $Q_1 = Q_2$, which is the case when all other tested detectors (SRNet and JRM) perform rather poorly.

Chapter 8

Revisiting Perturbed Quantization

In this work, we revisit Perturbed Quantization steganography with modern tools available to the steganographer today, including near-optimal ternary coding and content-adaptive embedding with side-information. In PQ, side-information in the form of rounding errors is manufactured by recompressing a JPEG image with a judiciously selected quality factor. This side-information, however, cannot be used in the same fashion as in conventional side-informed schemes nowadays as this leads to highly detectable embedding. As a remedy, we utilize the steganographic Fisher information to allocate the payload among DCT modes. In particular, we show that the embedding should not be constrained to contributing coefficients only as in the original PQ but should be expanded to the so-called "contributing DCT modes." This approach is extended to color images by slightly modifying the SI-UNIWARD algorithm. Using the best detectors currently available, it is shown that by manufacturing side information with double compression, one can embed the same amount of information into the doubly-compressed cover image with a significantly better security than applying J-UNIWARD directly in the single-compressed image. At the end of the paper, we show that double compression with the same quality makes side-informed steganography extremely detectable and should be avoided.

8.1 Introduction

Side-informed steganographic schemes are among the most secure steganographic schemes in existence today. The side-information typically comes in the form of rounding errors after some information-reducing processing applied to the (pre)cover image. One such processing is JPEG compression, which is known to provide high levels of security [31, 44, 77, 59, 60, 9, 51, 76, 75, 74]. The biggest drawback is that the steganographer needs to have access to the uncompressed image, considering that most imaging devices output images that are already compressed. The embedding method known as Perturbed Quantization (PQ) [51] manufactures side-information by recompressing the JPEG cover image in a way that maximizes the number of coefficients that fall in the middle of the quantization intervals during the second compression, and which are used for embedding. In this paper, we revisit this approach in light of modern tools presently available to the steganographer, such as content-adaptive embedding with costs modulated by the rounding errors [67, 24, 74] implemented using Syndrome Trellis Codes (STCs) [41] rather than the suboptimal wet paper codes [53] used in PQ. Additionally, due to the recent increased interest in embedding into color [121, 1, 2, 137, 22, 24], we extend the embedding to color JPEGs.

We do so while benchmarking the security with rich models [72, 96, 119, 30] and current state-of-the-art convolutional neural networks (CNNs) [8, 135, 130].

In Section 8.2, we introduce notation and describe the side-infor-mation produced by double compression. Section **??** explains the datasets and detectors used for evaluating the proposed method.

In Section 8.4, we derive a rule for selecting the second compression quality that provides, in some sense, the best side-information possible. The original PQ embedding is then modified to be able to embed larger payloads in images compressed with high qualities as well as in color images. Section **??** shows the experimental results on grayscale and color images. In Section 8.6, we delve into why double compression with the same quality should not be used as a source of side-information. The paper is concluded in Section 7.5.

8.2 Preliminaries and Notation

Boldface symbols are reserved for matrices and vectors with elementwise multiplication and division denoted $\odot$ and $\oslash$. Rounding x to the closest integer is denoted $[x]$. The set of all integers will be denoted $\mathbb{Z}$. For better readability, we strictly use i, j to index pixels and k, l DCT coefficients. Denoting by x_{ij}, $0 \leq i, j \leq 7$, an 8×8 block of pixels, they are transformed during JPEG compression to DCT coefficients $d_{kl} = \text{DCT}_{kl}(\mathbf{x}) \triangleq \sum_{i,j=0}^{7} f_{kl}^{ij} x_{ij}$, $0 \leq k, l \leq 7$, and then quantized $c_{kl} = [d_{kl}/q_{kl}]$, $c_{kl} \in \{-1024, \dots, 1023\}$, where q_{kl} are quantization steps in a luminance quantization matrix, and $f_{kl}^{ij} = w_k w_l / 4 \cos \pi k(2i+1)/16 \cos \pi l(2j+1)/16$, $w_0 = 1/\sqrt{2}$, $w_k = 1$, $0 < k \leq 7$, are the discrete cosines.

During decompression, the above steps are reversed. For a block of quantized DCTs c_{kl}, the corresponding block of non-rounded pixels after decompression is $y_{ij} = \text{DCT}_{ij}^{-1}(\mathbf{c} \odot \mathbf{q}) \triangleq \sum_{k,l=0}^{7} f_{kl}^{ij} q_{kl} c_{kl}$, $y_{ij} \in \mathbb{R}$. To obtain the final decompressed image, y_{ij} are rounded to integers $x_{ij} = [y_{ij}]$ and clipped to $[0, 255]$.

For compression of color images, the RGB representation is typically changed to YC_bC_r (luminance, and two chrominance signals) with:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ C_b &= 128 - 0.169R - 0.331G + 0.5B \\ C_r &= 128 + 0.5R - 0.419G - 0.081B \end{aligned} \tag{8.2.1}$$

The luminance channel Y is processed as described above, while the chrominance signals are optionally subsampled, then transformed using DCT, and finally quantized with chrominance quantization matrices [108]. In this work we avoid subsampling of chrominance signals because its effect on steganography has not been thoroughly studied yet.

8.2.1 Double Compression and Side Information

This work deals with embedding in JPEG images recompressed with a potentially different quality. The abbreviation SC will stand for single compressed and DC for double compressed images. To distinguish between DCT blocks and pixels of SC and DC images, we will use a superscript to keep track of the number of compressions. The symbol $\mathbf{c}^{(1)}$ represents the DCT block after the first compression, while $\mathbf{c}^{(2)}$ is the DCT block after the second compression. Similarly, $\mathbf{q}^{(1)}$ and $\mathbf{q}^{(2)}$ stand for quantization matrices in the first and second compression, respectively.

To obtain a DC image, a DCT block from the SC image, $\mathbf{c}^{(1)}$, is decompressed into $\mathbf{y}^{(1)} = \text{DCT}(\mathbf{c}^{(1)} \odot \mathbf{q}^{(1)})$, and rounded to integers $\mathbf{x}^{(1)} = [\mathbf{y}^{(1)}]$. We then compress with the second quantization table to obtain the DCT coefficients before quantization $\mathbf{d}^{(2)} = \text{DCT}(\mathbf{x}^{(1)})$. The final DCT coefficients after quantization are $\mathbf{c}^{(2)} = [\tilde{\mathbf{c}}^{(2)}] = [\mathbf{d}^{(2)} \oslash \mathbf{q}^{(2)}]$, where $\tilde{\mathbf{c}}^{(2)}$ are the quantized DCT coefficients before rounding to integers. Finally, the side-information created by recompression are the rounding errors during the last quantization $\mathbf{e} = \tilde{\mathbf{c}}^{(2)} - \mathbf{c}^{(2)}$.

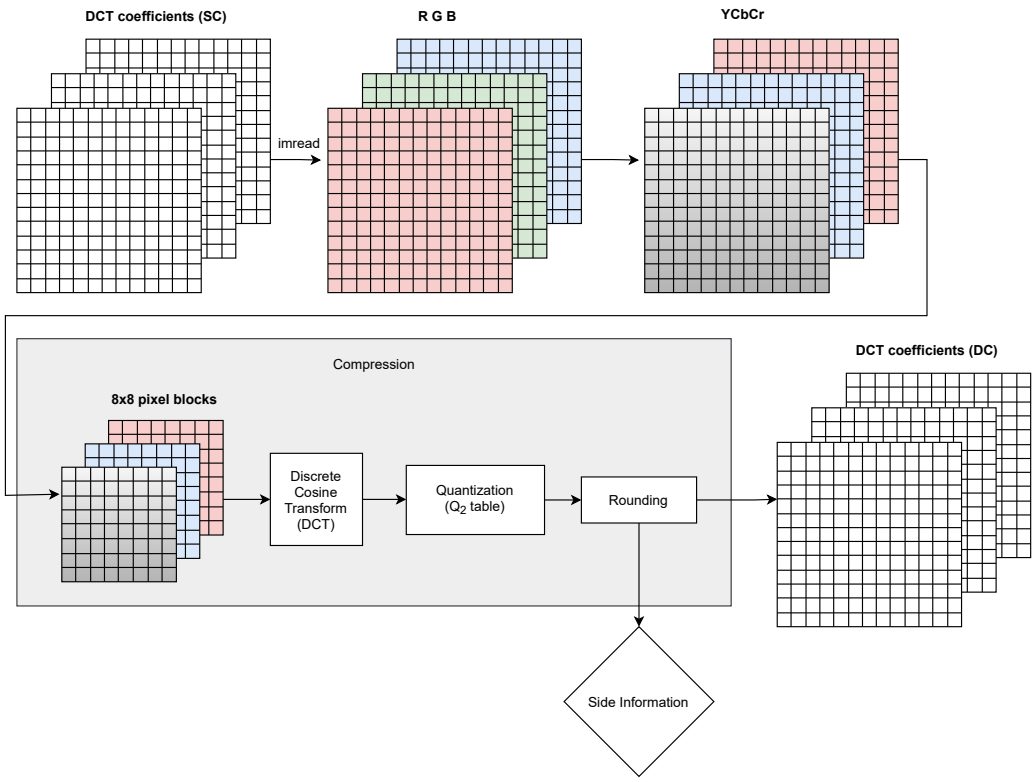

Figure 8.2.1: Double compression pipeline. We start with DCT coefficients of a single compressed (SC) image and end up with DCTs of a double compressed (DC) image.

To utilize these rounding errors for embedding, we follow the idea in [31] where the (symmetric) embedding costs ρ_{kl} of changing a DCT coefficient $c^{(2)}_{kl}$ by $+1$ or -1 are modulated by the rounding errors:

$$\begin{aligned} \rho_{kl}(\text{sign}(e_{kl})) &= (1 - |2e_{kl}|)\rho_{kl} \\ \rho_{kl}(-\text{sign}(e_{kl})) &= \rho_{kl}. \end{aligned} \tag{8.2.2}$$

8.3 Experimental Setup

This section describes the datasets as well as the detectors used for evaluating security.

8.3.1 Datasets

We work with two datasets to cover both grayscale and color images. The first dataset is a union of the popular BOSSbase 1.01 [4] and BOWS2 [5], each containing $10,000$ grayscale images downsampled to 256×256 using *imresize* with default parameters in Matlab. This union was then randomly split into training, validation, and testing sets with $14,000$, $1,000$, and $5,000$ images, respectively. This dataset was JPEG compressed with Matlab's *imwrite* with several quality factors Q_1. The second dataset is ALASKA 2 [23] consisting of three qualities 75, 90, 95, each having $25,000$ color images of size 512×512. This dataset was recently used in ALASKA II Kaggle competition.[1]

The compressed images represent the SC cover images (precovers) in our experiments. To obtain the DC cover images, the SC images are loaded into the *RGB* representation with Matlab's *imread*, converted into the YC_bC_r space via (8.2.1) (grayscale images are already loaded as Y channel), rounded to integers, and further compressed with quality Q_2 'manually' using Matlab's *dct2*. This was done in order to obtain the rounding errors $\mathbf{e}$ for the subsequent side-informed embedding. The resulting DCT coefficients were finally rounded to the nearest integers to obtain the DC cover images. As mentioned previously, we never used chrominance subsampling during compression of color images. This development pipeline is visualized in Figure 8.2.1.

We use the steganographic algorithm J-UNIWARD [74] for SC images as it is still one of the most secure algorithms for the JPEG domain in grayscale and color images when the development pipeline is not available [24, 20, 138]. For DC images, we use the side-informed version SI-UNIWARD [31] with several modifications, specific to DC images, as explained in the next section.

All experiments are set up in such a way that we always embed the same absolute payload size (in bits) in the SC image as in the DC image in order to answer the main question of this paper: "Can we embed the same amount of information more securely by recompressing the cover image?" The payload size will be expressed in bits per non-zero AC DCT coefficients (bpnzac) of SC cover image. All embedding algorithms are simulated on their corresponding rate–distortion bound (e. g., assuming optimal coding).

8.3.2 Detectors

Inspired by the fact that the best detectors in the recent ALASKA II Kaggle competition were mostly from the EfficientNet family, we attempted to train EfficientNet-B0 and B2 [105] on color images. However, these networks would not converge on the proposed DC steganographic scheme even after trying several different training schedules. Thus, in our experiments we used the SRNet [8] and rich models.

[1] https://www.kaggle.com/c/alaska2-image-steganalysis

Training the SRNet from scratch, however, was also impossible on the payloads used in this paper. There are many possible ways how to alleviate problems with convergence of a CNN detector. One can for example train on larger payloads first and use transfer learning on smaller payloads [10, 107, 143]. Alternatively, one can train on an 'easier' JPEG quality [12] or train on steganography in SC images first. To avoid confusion with so many different possibilities, we selected JIN[2] pretraining [**?**], which consists of pretraining on the ImageNet database [36] embedded with J-UNIWARD with uniform random payload between 0.4 and 0.6 bpnzac. This kind of pretraining is suitable for detecting steganography across a variety of embedding schemes embedding both in the JPEG and spatial domain, and for side-informed schemes [**?**]. All networks used for evaluation in this paper, for SC as well as DC images, were pretrained in this way. Since JIN pretraining is executed on color images, the networks pretrained in this way expect three-channel inputs. Thus, for grayscale images we simply replicated the grayscale representation in all three *RGB* channels. The network detectors were trained for 100 epochs in total on both datasets using mixed precision training with 64 images in every mini-batch, AdaMax optimizer, and weight decay 2×10^{-4}. We used OneCycle learning rate (LR) scheduler with maximum LR 10^{-3} at epoch 5, division factor 25 and final division factor 10. For easy implementation, PyTorch Lightning[3] framework was used for training our model. For DC images in BOSSbase+BOWS2 database embedded with 0.4 bpnzac, the pair constraint (PC) – forcing cover and its stego version in the same minibatch – was used for the first 50 epochs, otherwise the network would not converge even with the JIN pretraining. For every lower payload (in both datasets), transfer learning from 0.4 bpnzac was used without the PC for 50 epochs only.

For the rich models, we selected the ccJRM [94] and DCTR [72] feature sets with the ensemble classifier [97]. In color images, we use the JRM [96] instead of the cartesian-calibrated [94] ccJRM in order to keep a "manageable dimensionality" – the concatenation of extracted features from all three channels would triple the dimensionality of every feature set.

8.4 Perturbed Quantization

In this section, we review some concepts and basic facts from the original PQ method, such as the notion of a "contributing mode" and "contributing DCT coefficient", and justify the selection of the second quality factor for side-informed embedding in recompressed images.

Because double compression can introduce strong artifacts into the distribution of DCT coefficients [29, 142], it is important to avoid such combinations in steganography because the embedding could be very detectable using, e. g., the JPEG Rich Model (JRM). Figure 8.3.1 shows a few examples of artifacts due to double compression. In PQ [51], and in this paper as well, we wish to have after the second compression as many DCT coefficients with rounding errors $|e_{kl}| \sim 1/2$ as possible as the rounding of such coefficients can be intuitively perturbed with little impact on detectability. Note that this is in line with the modern understanding of side-informed steganography [31].

When recompressing a JPEG image compressed with quantization table $\mathbf{q}^{(1)}$ with quantization table $\mathbf{q}^{(2)}$, the DCT mode (k, l), $k, l = 0, \ldots, 7$ is called *contributing* if there exist $m, n \in \mathbb{Z}$ such that

$$m \cdot q_{kl}^{(1)} = n \cdot q_{kl}^{(2)} + \frac{1}{2} q_{k,l}^{(2)}. \tag{8.4.1}$$

These modes guarantee the existence of DCT coefficients with the absolute value of the rounding error in the DCT domain close to 1/2. As shown in [15], after recompression the DCT coefficients before rounding to integers follow a Gaussian distribution

$$\tilde{c}_{kl}^{(2)} \sim \mathcal{N}\left(c_{kl}^{(1)} \frac{q_{kl}^{(1)}}{q_{kl}^{(2)}}, \frac{1}{12 (q_{kl}^{(2)})^2} \right), \tag{8.4.2}$$

[2] JIN stands for **J**-UNIWARD embedded **I**mage**N**et
[3] https://www.pytorchlightning.ai/

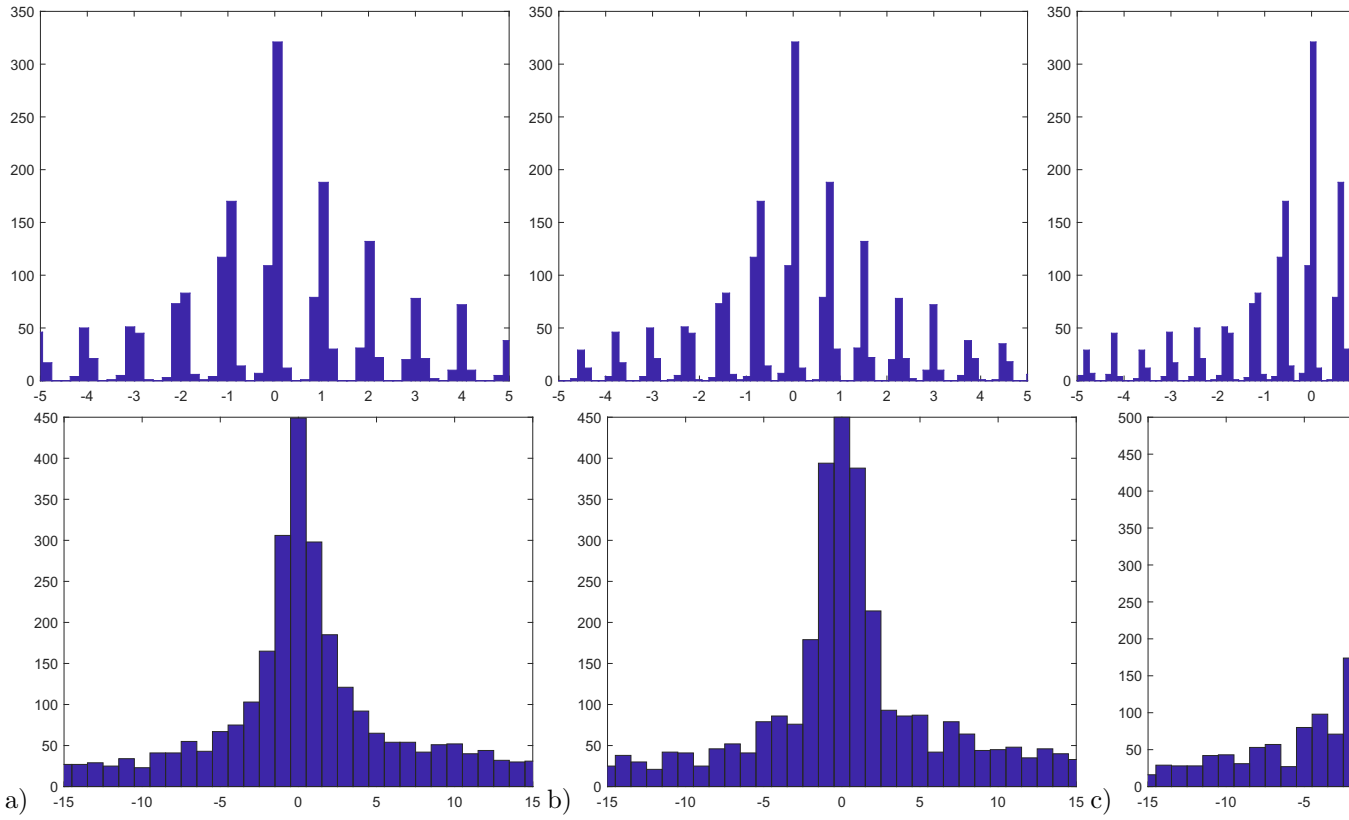

Figure 8.3.1: Histogram of a DCT mode compressed first with quantization step $q_{kl}^{(1)} = 3$ and further compressed with quantization step $q_{kl}^{(2)}$ equal to a) 3, b) 4, c) 5, d) 6. Top: before rounding of the DCT coefficients, bottom: after rounding. The spikes in top row are around multiples of $q_{kl}^{(1)}/q_{kl}^{(2)}$. Only cases b) and d) correspond to contributing modes.

where the mean can be written from (8.4.1) as

$$\begin{aligned}\mathbb{E}[\tilde{c}_{kl}^{(2)}] &= c_{kl}^{(1)}\frac{q_{kl}^{(1)}}{q_{kl}^{(2)}} \\ &= n' + \frac{1}{2},\end{aligned} \tag{8.4.3}$$

where it was assumed that $c_{kl}^{(1)}$ and some $n' \in \mathbb{Z}$ play the role of m, n in (8.4.1). With such a small variance (8.4.2), it follows that after rounding to the nearest integers the rounding errors of these coefficients will be clustered around $\pm 1/2$.

In [51], the following useful theorem is proved.

Theorem 1. *The mode (k,l) is contributing if and only if $q_{kl}^{(2)}/g$ is even, where $g = \gcd(q_{kl}^{(1)}, q_{kl}^{(2)})$ is the greatest common divisor of $q_{kl}^{(1)}$ and $q_{kl}^{(2)}$. Furthermore, all contributing multiples m of $q_{kl}^{(1)}$ are expressed by the formula*

$$m = (2n+1)\frac{q_{kl}^{(2)}}{2g},\ n \in \mathbb{Z}. \tag{8.4.4}$$

In PQ steganography, embedding is executed only in contributing coefficients, which by Theorem 1, means in coefficients satisfying $c_{kl}^{(1)} = (2n+1)\frac{q_{kl}^{(2)}}{2g}$ for some $n \in \mathbb{Z}$. We wish to emphasize that not all coefficients in contributing modes are contributing.

The motivation behind using only these coefficients is simple. It was shown [15] that the rounding errors in the DCT domain after the second compression e_{kl} follow a Gaussian distribution folded into the interval $[-1/2, 1/2]$:

$$e_{kl} \sim \mathcal{N}_F\left(c_{kl}^{(1)}\frac{q_{kl}^{(1)}}{q_{kl}^{(2)}}, \frac{1}{12(q_{kl}^{(2)})^2}\right), \tag{8.4.5}$$

where the mean of this distribution is $\mathbb{E}[e_{kl}] = c_{kl}^{(1)}\frac{q_{kl}^{(1)}}{q_{kl}^{(2)}} - [c_{kl}^{(1)}\frac{q_{kl}^{(1)}}{q_{kl}^{(2)}}]$. It is then clear that for a contributing mode (k,l)

$$\begin{aligned}\mathbb{E}[e_{kl}] &= c_{kl}^{(1)}\frac{q_{kl}^{(1)}}{q_{kl}^{(2)}} - [c_{kl}^{(1)}\frac{q_{kl}^{(1)}}{q_{kl}^{(2)}}] \\ &= c_{kl}^{(1)}\frac{q_{kl}^{(1)}/g}{q_{kl}^{(2)}/g} - [c_{kl}^{(1)}\frac{q_{kl}^{(1)}/g}{q_{kl}^{(2)}/g}] \\ &= c_{kl}^{(1)}\frac{u}{2v} - [c_{kl}^{(1)}\frac{u}{2v}]\end{aligned} \tag{8.4.6}$$

for some $u, v \in \mathbb{Z}$ coprime because Theorem 1 states that the denominators in (8.4.6) are even. Then in every case where v divides $c_{kl}^{(1)} \cdot u$ and $2v$ does not divide $c_{kl}^{(1)} \cdot u$ we get the desirable $|\mathbb{E}[e_{kl}]| = 1/2$.

Equipped with this knowledge, we would now like to maximize the number of rounding errors that are close to $1/2$ in absolute value. Because the coefficients of the SC image $c_{kl}^{(1)}$ are given, the easiest way to ensure this for as many coefficients as possible is to let v divide u. Since u, v are coprime, this means $u = v = 1$ and thus $q_{kl}^{(2)} = 2q_{kl}^{(1)}$.[4] In this case, a coefficient $c_{kl}^{(1)}$ from a contributing mode is contributing whenever it is odd.

[4] This relationship was derived in [51] only experimentally by virtue of Figure 3 in Sec. 4.3.

Enforcing the constraint $q_{kl}^{(2)} = 2q_{kl}^{(1)}$, however, would lead to non-standard quantization tables, and thus potentially an easy artifact of embedding. This is why in our work, we limit ourselves to standard quantization tables. Recall that the luminance quantization table for quality factor Q is defined as

$$\mathbf{q}(Q) = \begin{cases} \max\left\{\mathbf{1}, \left[2\mathbf{q}(50)\left(1 - \frac{Q}{100}\right)\right]\right\}, & Q > 50 \\ \min\left\{255 \times \mathbf{1}, \left[\mathbf{q}(50)\frac{50}{Q}\right]\right\}, & Q \leq 50, \end{cases} \tag{8.4.7}$$

where the luminance quantization table for quality factor 50 is

$$\mathbf{q}(50) = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}. \tag{8.4.8}$$

For the chrominance quantization table $\mathbf{q}_C(Q)$ at quality Q, the same formula (8.4.7) applies with chrominance quantization table at quality 50

$$\mathbf{q}_C(50) = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}. \tag{8.4.9}$$

For simplicity, let us now work only with luminance quantization tables and $Q > 50$

$$\mathbf{q}(Q) = 2\mathbf{q}(50)\left(1 - \frac{Q}{100}\right). \tag{8.4.10}$$

Combining with our condition $q_{kl}^{(2)} = 2q_{kl}^{(1)}$, we obtain a relationship between the first and second quality factors Q_1 and Q_2:

$$\begin{aligned} \mathbf{q}(Q_2) &= 2\mathbf{q}(Q_1) \\ &= 2\left(2\mathbf{q}(50)\left(1 - \frac{Q_1}{100}\right)\right) \\ &= 2\mathbf{q}(50)\left(1 - \frac{2(Q_1 - 50)}{100}\right) \\ &= \mathbf{q}(2(Q_1 - 50)) \end{aligned}$$

or

$$Q_2 = 2(Q_1 - 50), \tag{8.4.11}$$

as also reported in [51] based on experiments. In this work, we will follow this recipe for the selection of Q_2 with one exception for $Q_1 = 100$, because in this case we would declare $Q_2 =$

Q	(75,50)	(90,80)	(95,90)
SI all - binary	0.0915	0.0222	0.0091
SI all - ternary	0.0957	0.0250	0.0155
J-UNIWARD	0.2777	0.3599	0.3932

Table 8.1: P_{E} with DCTR at 0.4 bpnzac of J-UNIWARD in single compressed images, and SI-UNIWARD in double compressed images while embedding into all DCT modes, binary and ternary version. BOSSbase+BOWS2 dataset.

100 and the embedding would be reliably detected using the Reverse JPEG Compatibility Attack (RJCA) [13, 21]. For this reason, for $Q_1 = 100$, we heuristically choose $Q_2 = 98$ as the largest quality not attackable by the RJCA.

Additionally, the same relationship holds for the chrominance quantization tables, which will help us extend this idea to color images. To relax the notation, from now we denote $Q = (Q_1, Q_2)$ the pair of quality factors used for recompression with Q_1 used for SC images.

8.4.1 Naive application of side-information

The most straightforward way to cast the idea behind the PQ within the modern embedding paradigm is to use a modern content-adaptive steganographic method, such as J-UNIWARD, and apply the standard way of incorporating side-information by modulating the embedding costs by the rounding errors obtained during recompression (8.2.2). Table 8.1 shows the comparison of such SI-UNIWARD scheme in DC images with J-UNIWARD in SC images under the assumption that the exact same absolute payload is embedded by both schemes. The side-informed scheme is much more detectable than non-informed J-UNIWARD in SC images. To make sure that the high detectability is not introduced by ternary embedding, we also include the results for the binary version of SI-UNIWARD. Both the binary and ternary versions, however, exhibit a similar level of (in)security.

We now investigate where this high detectability comes from. We will measure the impact of the embedding on the distribution of DCT coefficients from every mode (k, l) using the steganographic Fisher Information

$$I_{kl} = \sum_{m\in\mathbb{Z}} \frac{1}{p^{(c)}_{kl}(m)} \left(\frac{\partial p^{(s)}_{kl}(m)}{\partial \alpha}|_{\alpha=0} \right)^2, \tag{8.4.12}$$

where $p^{(c)}_{kl}$ is the cover probability mass function (pmf) of DCT coefficients in mode (k, l), $p^{(s)}_{kl}$ is the pmf of stego images in the same mode, and α is the relative payload size. Since we cannot easily model the stego pmf when using J-UNIWARD, we approximate the Fisher information with real data as

$$\tilde{I}_{kl} = \sum_{m\in\mathbb{Z}} \frac{1}{h^{(c)}_{kl}(m)} \left(\frac{h^{(c)}_{kl}(m) - h^{(s)}_{kl}(m)}{\alpha} \right)^2, \tag{8.4.13}$$

where we use the actual histograms $h^{(c)}_{kl}$ and $h^{(s)}_{kl}$ of the cover and the corresponding stego images embedded with relative payload α. We average (8.4.13) over 100 randomly chosen images from the BOSSbase dataset and show in Figure 8.4.1 the average FI per mode together with the contributing modes for three different qualities Q. We used payload $\alpha = 1.1$ bpnzac for the embedding of stego images because for smaller payloads the approximation of the FI (8.4.13) does not utilize many changes in histograms and thus does not provide any useful feedback. We can clearly see

Payload	0.3 bpnzac			0.4 bpnzac			0.5 bpnzac			0.6 b	
Q	(75,50)	(90,80)	(95,90)	(75,50)	(90,80)	(95,90)	(75,50)	(90,80)	(95,90)	(75,50)	(90
Binary contr coefficients	0.4082	0.3871	0.4197	0.3424	0.3381	0.0164	0.1990	0.0385	0.0017	0.0230	0.0
Binary, contr modes	0.4085	0.3895	0.4477	0.3441	0.3526	0.2940	0.2705	0.2813	0.0167	0.2118	0.1
Ternary, contr modes	0.4034	0.3922	0.4536	0.3660	0.3587	0.3530	0.2909	0.3118	0.1929	0.2375	0.2

Table 8.2: P_{E} with DCTR of SI-UNIWARD in double compressed images. Comparison between embedding into contributing coefficients and all coefficients in contributing modes. Binary and ternary embedding. BOSSbase+BOWS2 dataset.

a relationship between the non-contributing modes and the modes with high $\tilde{I}_{kl}$, which suggests that embedding in these modes is much more detectable. The only notable exception to this is in high frequencies of the lowest tested quality $Q = (75, 50)$. In this case, almost all cover coefficients are equal to zero due to the strong quantization, which leads to inaccurate estimates of the Fisher Information. We further report that the average FI across all non-contributing modes is 2–5 times larger than the average FI in the contributing modes. Remembering that the FI is in the error exponent of the likelihood ratio test, allowing embedding changes in non-contributing modes will have a grave impact on security.

To further support that the embedding into non-contributing modes is the culprit, we show in Figure 8.4.2 boxplots of the differences between stego and cover histograms. The differences in histograms exist because of the bias in the SI embedding towards coefficients with large rounding errors due to the nature of the cost modulation (8.2.2). For non-contributing modes, these coefficients are located at the peaks of mode histograms (see Figure 8.3.1 c)), which after embedding causes a very detectable distortion in the DCT mode histogram because these peaks will get deformed. Contributing modes do not suffer from this because they either have double peaks in histograms, which will be preserved during embedding, or no peaks (except at zero) (see Figure 8.3.1 b) and d)).

8.4.2 Restricting the embedding

The results from the previous section give a direction on how to adjust the side-informed embedding in double compressed images in order to avoid introducing changes into structures that exist in the distribution of coefficients of DC images. Constraining the embedding only to contributing multiples of $q_{kl}^{(1)}$ (8.4.4) as in the original PQ algorithm seems like the best option, however, this severely limits the capacity of the embedding. Table 8.2 shows the detection error with DCTR features across a wide range of payloads. Once the payload reaches 0.4 bpnzac at $Q = (95, 90)$, the detection error drops drastically. We verified that these drops indeed correspond to embedding messages that are simply too large to fit only into contributing coefficients. Hence, the embedding algorithm starts making changes in other coefficients, which happens without any content-adaptivity because the embedding spills into forbidden coefficients assigned with the same "wet cost."

Since we cannot embed into all modes securely and embedding only into contributing coefficients seems very limiting in terms of the maximal embeddable payload, we consider embedding into all coefficients from contributing modes because of the smaller impact of the embedding in terms of the FI (8.4.13) (see Figure 8.4.1 for an example).

Figure 8.4.3 shows the embedding capacity as the number of "changeable coefficients" per non-zero AC DCT coefficients of the single compressed image. Changeable coefficients are either only contributing coefficients or all coefficients from all contributing modes. It was verified for a range of qualities that for grayscale images, using all coefficients in contributing modes increases the average embedding capacity by approximately 50%. An even more pronounced effect can be observed for color images, where the number of contributing coefficients is bimodal (Figure 8.4.3 bottom left) caused by images that have very little or no contributing coefficients in chrominance channels. Such

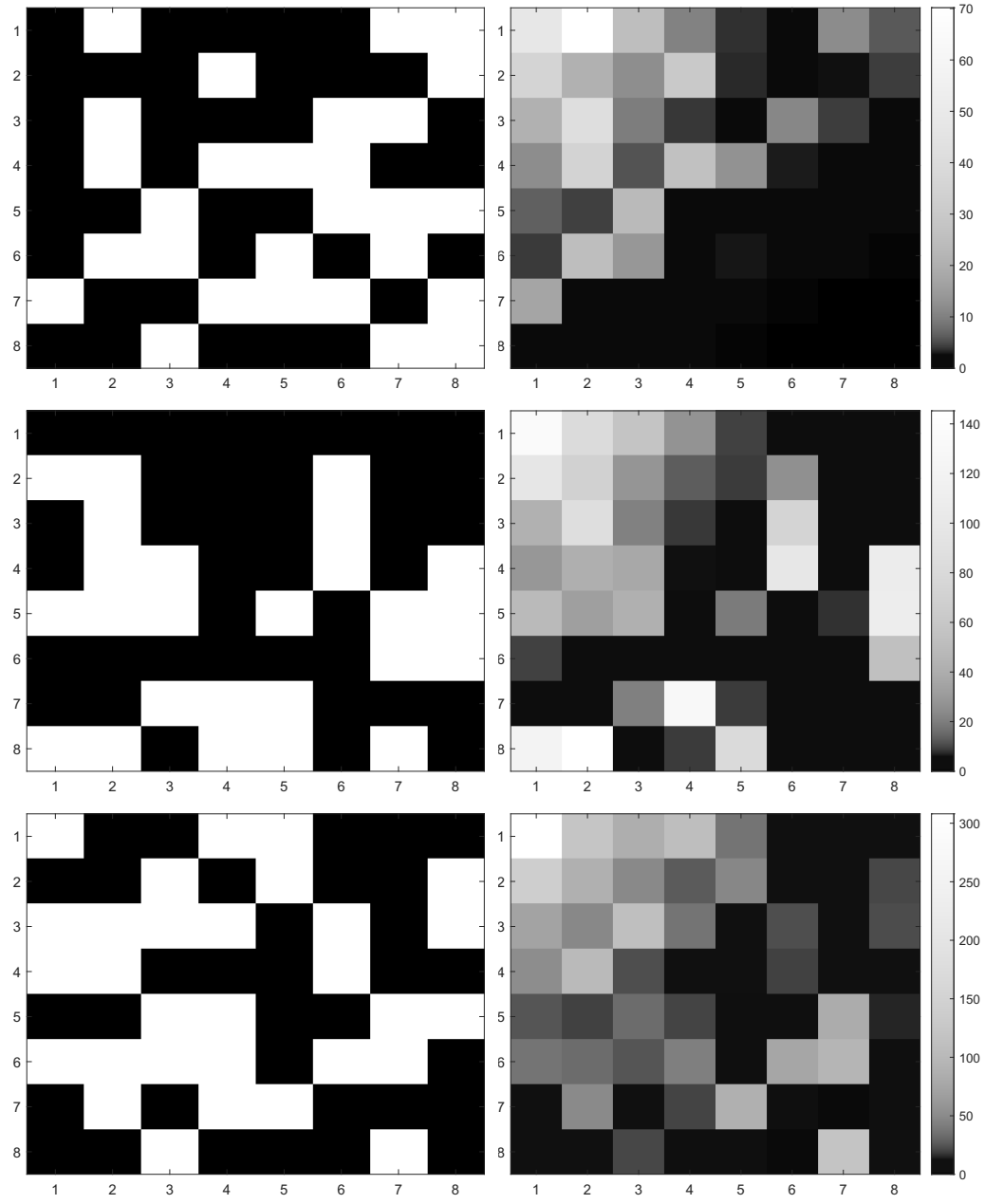

Figure 8.4.1: Top: $Q = (75, 50)$, middle: $Q = (90, 80)$, bottom: $Q = (95, 90)$. Left: in black are contributing DCT modes, in white are non-contributing modes. Right: approximation of FI $\tilde{I}_{kl}$ per mode averaged over 100 images from BOSSbase embedded with 1.1 bpnzac.

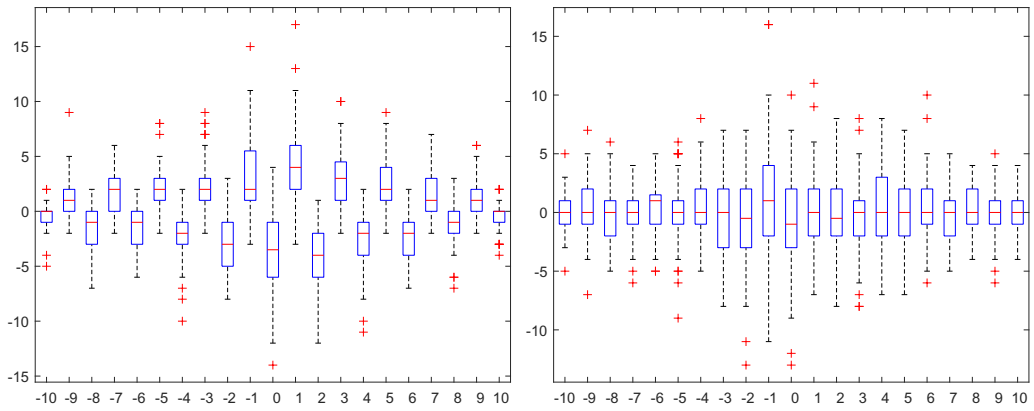

Figure 8.4.2: Boxplots showing the differences between the distribution of DCT coefficients from stego images embedded with SI-UNIWARD (0.4 bpnzac) when embedding into all modes and cover images across 100 randomly selected images from BOSSbase with double compression quality $Q = (90, 80)$. Left: non-contributing mode $(2, 1)$ with quantization steps 2 and 5, Right: contributing mode $(1, 2)$ with quantization steps 2 and 4.

images are not uncommon, thanks to the nature of the chrominance quantization table (8.4.9). The distribution of the number of all coefficients from contributing modes is better behaved (Figure 8.4.3 bottom right).

For a larger embedding capacity, we therefore relax the embedding restriction by allowing embedding into all coefficients inside contributing modes, not only the contributing coefficients. The results are shown in Table 8.2, where we can see that for payloads as large as 0.6 bpnzac, the ternary embedding into all coefficients inside contributing modes provides overall best security. Based on this analysis, we will keep using this embedding strategy for the rest of the paper.

8.4.2.1 High qualities

In the derivation of (8.4.11), we did not consider the nonlinear dependence of quantization steps on the quality factor due to taking the maximum with one and rounding. While the rounding operation introduces the same nonlinearity for every quantization step regardless of the quality factor applied, the maximum will only be applied for very high quality factors and mainly for low frequency modes. Note that if $Q_2 = 2(Q_1 - 50)$, then $q_{kl}^{(1)} = q_{kl}^{(2)}$ if and only if $q_{kl}^{(2)} = 1$.This introduces an issue that needs to be addressed, because when the maximum starts introducing ones in the second quantization table (this occurs for $Q_2 \geq 93$), we would end up, with our definition of a contributing mode, with very few contributing modes. This is because if a second quantization step is equal to one, then $q_{kl}^{(2)}/\gcd(q_{kl}^{(1)}, q_{kl}^{(2)}) = 1$ is not even, which would effectively prevent us from embedding non-trivial payloads. To this end, we decided to allow embedding into modes with $q_{kl}^{(1)} = q_{kl}^{(2)}$. Such modes are not contributing, but the embedding does not suffer from these modes since this combination of quantization steps does not introduce easily exploitable artifacts (the JRM performs very poorly in these cases [15]). Figure 8.3.1 a) also suggests that the histogram of such modes does not start showing any drastic artifacts. Even though the mean of the DCT error (8.4.6) is zero in these cases, its variance (8.4.5) is equal to $1/12$, which still ensures quite a few of the DCT rounding errors to be close to $\pm 1/2$. The effect of allowing embedding in these modes can be seen in Figure 8.4.4. We verified that the high detectability for the case where we do not allow embedding into modes with $q_{kl}^{(1)} = q_{kl}^{(2)}$ comes from the payload being too large, a problem we have encountered in the previous section too, while trying to embed only into contributing coefficients. Consequently, the embedding changes were made in non-contributing modes without content-adaptivity.

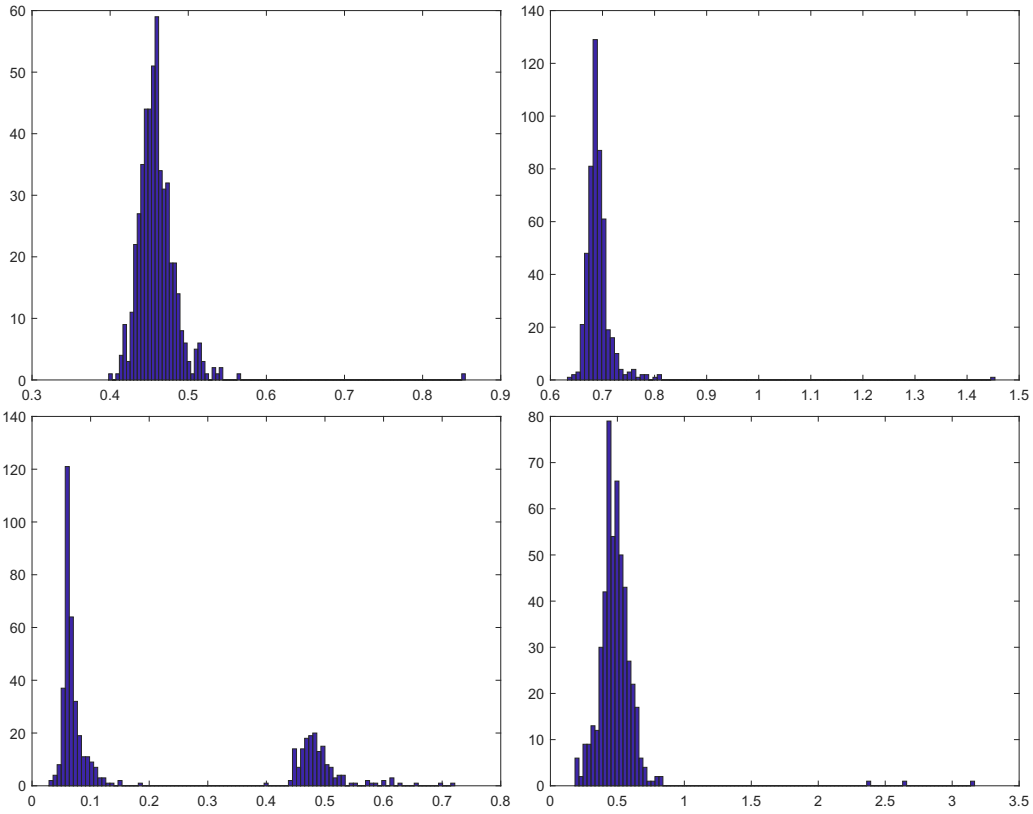

Figure 8.4.3: Average number of changeable coefficients per non-zero AC DCT coefficients over 500 randomly chosen images. Top: BOSSbase+BOWS2 (grayscale), bottom: ALASKA 2 (color), left: embedding only into contributing coefficients, right: embedding into all coefficients in contributing modes.

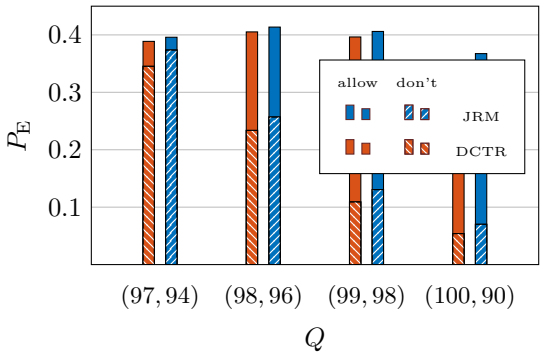

Figure 8.4.4: Detection error P_E of SI-UNIWARD at 0.4 bpnzac in DC images when modes with $q_{kl}^{(1)} = q_{kl}^{(2)}$ are/are not allowed for embedding. BOSSbase+BOWS2 dataset.

8.4.2.2 Color

Embedding in color images can spread the payload across luminance and the two chrominance channels. Several different payload spreading strategies into the three YC_bC_r channels were recently proposed in [24, 121]. It was reported in [24] that for J-UNIWARD, the CCM (Color Channels Merging), which distributes the payload by minimizing the additive distortion across all three channels, and CCFR (Color Channels Fixed Repartition) with repartition parameter $\gamma = 0.2$, which puts a fraction of payload into chrominance channels, provide almost the same level of security. We wanted to verify whether this remains true for SI-UNIWARD in DC images. After testing with DCTR on SI-UNIWARD with CCM and CCFR(0.2), we found, to our surprise, that the CCM strategy was much more detectable. We believe CCM should be the optimal strategy for spreading the payload because it distributes the payload automatically without forcing a fixed portion of the payload into chrominance. It was identified that the poor performance of CCM is caused by the discrepancy between the embedding costs in luminance and chrominance channels, which forces a vast majority of the payload into the luminance channel. After careful inspection of the embedding algorithm for SI-UNIWARD, we realized that the culprit was the stabilizing constant σ used in J-UNIWARD's distortion function [74]:

$$D(\mathbf{X},\mathbf{Y}) = \sum_{k=1}^{3}\sum_{u=1}^{n_1}\sum_{v=1}^{n_2} \frac{|W_{uv}^{(k)}(\mathbf{X}) - W_{uv}^{(k)}(\mathbf{Y})|}{\sigma + |W_{uv}^{(k)}(\mathbf{X})|}, \tag{8.4.14}$$

where $\mathbf{X}$ and $\mathbf{Y}$ represent the cover and stego images in the pixel domain (in one channel), n_1, n_2 are the number of DCT blocks in the vertical and horizontal directions, and $W_{uv}^{(k)}(\cdot)$ the wavelet transformation based on Daubechies 8-tap wavelet directional filter bank. By default, σ is set to 2^{-6}, which would not be an issue if the normalization factor in (8.4.14) was on a similar scale for luminance and chrominance channels. While this is true for SC images, for DC images it is not. In fact, $|W_{uv}^{(k)}(\mathbf{X})|, k \in \{1,2,3\}$ in chrominance channels can be by several orders of magnitude smaller than in the luminance channel. We believe that this is due to much harsher quantization in chrominance channels of DC images compared to SC images (see the quantization tables (8.4.8) and (8.4.9)). Thus, we claim that the stabilizing constant has to be smaller in chrominance channels. Keeping the original luminance stabilizing constant $\sigma_Y = 2^{-6}$, in Figure 8.4.5 we show P_{E} of DCTR on SI-UNIWARD with 0.4 bpnzac with the CCM spreading strategy across a range of values for the stabilizing constant in chrominance channels σ_C. We see that for qualities $(90, 80)$ and $(95, 90)$, σ_C is reaching the best security for $\sigma_C = 2^{-15}$. For the lowest quality $(75, 50)$, the most secure σ_C is at 2^{-16}. In order to have a unified setting, we declare $\sigma_C = 2^{-15}$ for every quality combination, even at a loss for the low qualities. With σ_C adjusted this way, we searched for optimal σ_Y. Coincidentally, the default value $\sigma_Y = 2^{-6}$ provides the best performance.

8.5 Evaluation

To show the benefit of embedding in recompressed images, we contrast the empirical security with embedding in the corresponding single-compressed cover images. To summarize the embedding algorithm, we use ternary embedding in all coefficients belonging to contributing modes and modes with $q_{kl}^{(1)} = q_{kl}^{(2)}$. For color images, we furthermore improved the security by changing the chrominance stabilizing constant σ_C of J-UNIWARD's costs. The second quality factor Q_2 used for recompression is selected by Eq. (8.4.11) with one exception for $Q_1 = 100$ where we set $Q_2 = 98$.

8.5.1 Grayscale

We test the proposed scheme on a range of qualities with the detectors described in Section **??**. We also tested GFR [119] and its selection channel aware version, where we used J-UNIWARD for

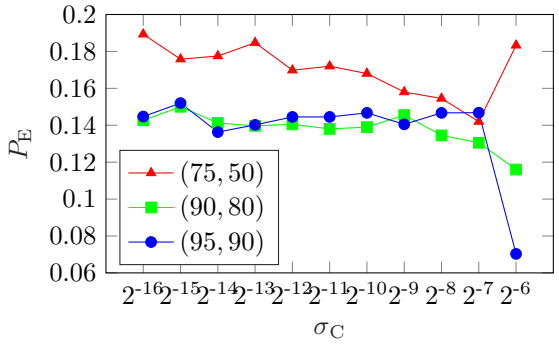

Figure 8.4.5: P_E with DCTR of CCM-SI-UNIWARD in DC images with different values of the stabilizing constant σ_C of chrominance channels C_r and C_b, with the luminance constant at the default $\sigma_Y = 2^{-6}$. Three qualities $(75, 50), (90, 80)$, and $(95, 90)$ are shown. ALASKA 2 dataset.

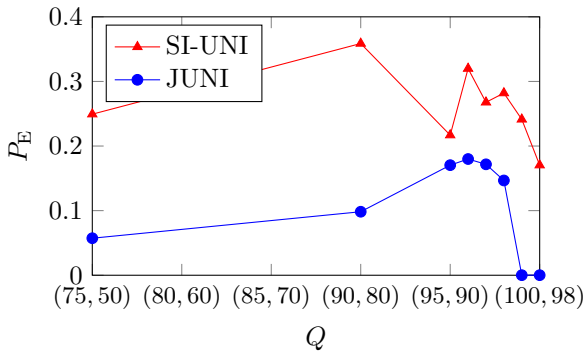

Figure 8.5.1: Detection error P_E of J-UNIWARD in SC images and SI-UNIWARD in DC images at 0.4 bpnzac. Only the best detector's performance is shown. BOSSbase+BOWS2 dataset.

Q	Detector	0.1 bpnzac		0.2 bpnzac		0.3 bpnzac		0.4 bpnzac	
		DC-SI	JUNI	DC-SI	JUNI	DC-SI	JUNI	DC-SI	JUNI
(75,50)	ccJRM	0.4505	0.4850	0.4439	0.4552	0.4225	0.4078	0.4147	0.3631
	DCTR	**0.4484**	0.4697	**0.4397**	0.4202	0.4034	0.3465	0.3660	0.2777
	SRNet	0.5000	0.3094	0.5000	0.1764	**0.3189**	0.0961	**0.2493**	0.0573
(90,80)	ccJRM	0.4134	0.4900	**0.4057**	0.4767	0.4020	0.4588	0.3849	0.4144
	DCTR	**0.4114**	0.4860	0.4061	0.4546	**0.3922**	0.4185	**0.3587**	0.3599
	SRNet	0.4469	0.3661	0.4461	0.2522	0.4390	0.1519	0.3884	0.0983
(95,90)	ccJRM	0.4876	0.4872	0.4881	0.4613	0.4736	0.4351	0.3771	0.3990
	DCTR	**0.4823**	0.4941	**0.4839**	0.4714	0.4536	0.4441	0.3530	0.3932
	SRNet	0.5000	0.4297	0.5000	0.3292	**0.3686**	0.2401	**0.2169**	0.1704

Table 8.3: Detection error P_{E} of SRNet, ccJRM, and DCTR for various payloads (bpnzac) of J-UNIWARD in SC and SI-UNIWARD in DC images. Boldface represents the best detector of the more secure algorithm at a fixed payload. BOSSbase+BOWS2 dataset.

estimating the selection channel. Both of these feature sets, however, did not bring any improvement over DCTR. For the highest qualities $(99, 98)$ and $(100, 98)$, we also trained e-SRNet [13], SRNet trained on rounding errors of pixel values after decompression, as it is the best detector for the highest quality JPEGs. Only the best detector's detection error P_{E} on SI-UNIWARD in DC images and J-UNIWARD in SC images with 0.4 bpnzac is shown in Figure 8.5.1. For J-UNIWARD, the best detector is always the SRNet, while for the two highest qualities, it is e-SRNet (note the extremely low errors). The best detector for SI-UNIWARD is also SRNet, with one exception at quality $(90, 80)$, where DCTR provides a better detection. The e-SRNet performed substantially worse than SRNet, confirming that the RJCA is not applicable with the quality selection rule (8.4.11). Overall, the improvement of embedding in recompressed images when compared to J-UNIWARD ranges between $5 - 25\%$ in terms of P_{E}.

To obtain a better understanding of how the algorithms compare for smaller payloads, we trained the SRNet, ccJRM, and DCTR at qualities $(75, 50), (90, 80)$, and $(95, 90)$ for various payloads. The results are shown in Table 8.3. We can see clear improvement over J-UNIWARD at every payload. Surprisingly, in many cases (especially for the lowest payloads), DCTR provides a better detection than SRNet on SI-UNIWARD. This suggests that the SRNet is not able to collect detection statistics from a somewhat detectable distortion in the DCT domain.

8.5.2 Color

Setting the chrominance stabilizing constant $\sigma_C = 2^{-15}$, we first reevaluate the spreading strategies CCFR and CCM [24]. Table 8.4 shows DCTR's P_{E} on the CCFR strategy for several values of the repartition parameter γ. With increasing quality, the optimal parameter γ also needs to grow as the best value of γ for every quality is different. Interestingly, CCFR strategy with $\gamma = 0.2$ outperforms CCM at quality $(75, 50)$, but on the other two tested qualities, CCM achieves a better security. In Table 8.5, we include a comparison between SI-UNIWARD in DC images with $\sigma_C = 2^{-15}$ and J-UNIWARD in SC images across several payloads and several qualities, both schemes using the CCM payload spreading strategy. Using side-information provides an improvement in security up to 18% at quality $(75, 50)$ and payload 0.2 bpnzac. Interestingly, the non-informed J-UNIWARD is more secure in two tested scenarios: $Q = (90, 80)$ at 0.1 bpnzac and $Q = (95, 90)$ at 0.4 bpnzac. The latter is most likely caused by the large embedding payload in DC images because, as can be seen in Figure 8.4.3, the embedding capacity in color images is lower than in grayscale images. This is in line with the significant jumps in P_{E} of SI-UNIWARD for lower payloads at $Q = (95, 90)$.

8.6 Double compression with the same quality

In this section, we investigate the case of side-informed steganography in images that were double compressed with the same quantization table. We included this analysis because the option $Q_1 = Q_2$

	Repartition parameter γ				
Q	0.1	0.2	0.3	0.4	0.5
(75,50)	0.1893	**0.2008**	0.1835	0.1517	0.1120
(90,80)	0.1105	0.1110	**0.1265**	0.1162	0.0772
(95,90)	0.0645	0.0837	0.1115	**0.1247**	0.0757

Table 8.4: P_{E} with DCTR of CCFR-SI-UNIWARD at 0.4 bpnzac in DC images with chrominance stabilizing constant $\sigma_C = 2^{-15}$. ALASKA 2 dataset.

Q	Detector	0.1 bpnzac		0.2 bpnzac		0.3 bpnzac		0.4 bpnzac	
		DC-SI	JUNI	DC-SI	JUNI	DC-SI	JUNI	DC-SI	JUNI
(75,50)	JRM	**0.3362**	0.4845	**0.2957**	0.4547	**0.2120**	0.4138	**0.1210**	0.3740
	DCTR	0.3708	0.4100	0.3478	0.2937	0.2735	0.1867	0.1758	0.1108
	SRNet	0.4093	0.2516	0.3243	0.1119	0.2736	0.0607	0.2524	0.0327
(90,80)	JRM	0.2885	0.4750	**0.2658**	0.4477	0.2368	0.4025	0.1903	0.3653
	DCTR	0.3120	0.4473	0.2835	0.3652	**0.2085**	0.2740	**0.1500**	0.1947
	SRNet	0.3978	**0.3473**	0.3933	0.2236	0.3353	0.1397	0.2394	0.0857
(95,90)	JRM	**0.4300**	0.4305	0.4088	0.3455	0.3310	0.2758	0.2208	0.2248
	DCTR	0.4305	0.4542	**0.4032**	0.3800	0.2883	0.3163	0.1520	0.2223
	SRNet	0.5000	0.4193	0.4268	0.3083	**0.2604**	0.2211	0.1372	**0.1524**

Table 8.5: P_{E} of SI-UNIWARD in DC images with $\sigma_C = 2^{-15}$ and J-UNIWARD in SC images, both using CCM strategy. ALASKA 2 dataset.

avoids introducing any histogram artifacts and it would allow us to embed into every DCT mode, thus significantly increasing the embedding capacity. Furthermore, and most importantly, it is not immediately obvious that side-informed embedding in this setup is extremely detectable and exhibits some very unusual properties, such as higher statistical detectability of smaller payloads than larger payloads.

As shown in [15], embedding in DC images with $Q_1 = Q_2$ can be attacked with the RJCA. However, this work did not investigate the case of side-informed embedding. Since everywhere in this section it is assumed that $Q_1 = Q_2$, we will again refer to the compression quality simply as Q.

The performance of the e-SRNet as implemented in [15] can be seen in Figure 8.6.1. Note that the detection errors are much lower than for quality factor rule (8.4.11) in Table 8.3. Moreover, the most peculiar behavior can be observed for $Q < 93$ when the detection of smaller payloads is more reliable. We will now show that the rounding errors $\mathbf{e}$ can actually be partly recovered from the double compressed (and embedded) images with $Q_1 = Q_2$, which is responsible for this peculiar behavior.

8.6.1 Estimating the side information

Let us call the changes in the DCT coefficients introduced during the second compression as *inconsistencies*. In other words, the compression produces an inconsistency at $c_{kl}^{(2)}$ if $c_{kl}^{(1)} \neq c_{kl}^{(2)}$. Figure 8.6.2 shows that for $Q < 93$ the second compression does not introduce many inconsistencies mainly because there are no ones in the quantization table. We hypothesize that for lower qualities (where quantization tables do not contain any ones, i. e., $Q < 93$) the following claim holds: the fewer inconsistencies the better the estimate of the rounding error $\mathbf{e}$ can be obtained. Intuitively, this makes sense because if the second compression does not change any coefficient in a given DCT block then also the third compression would not change any coefficients. Therefore, one can compute the rounding errors $\mathbf{e}$ used during embedding (and thus nullify the effect of side-information) by simply compressing the DC image once more. Note that the embedding changes can also be considered inconsistencies. If the claim holds, it would immediately mean that we can get a better estimate of $\mathbf{e}$ with decreasing payload.

To estimate the DCT errors $\mathbf{e}$, we decompress a given (double compressed and possibly embedded)

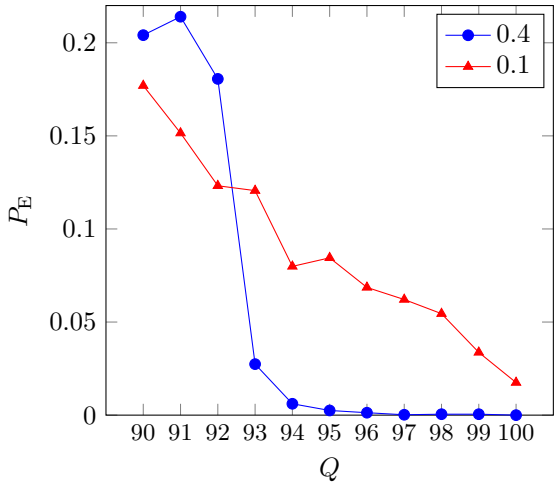

Figure 8.6.1: P_{E} with e-SRNet of SI-UNIWARD in DC images at 0.1 and 0.4 bpnzac when $Q_1 = Q_2$. BOSSbase+BOWS2 dataset.

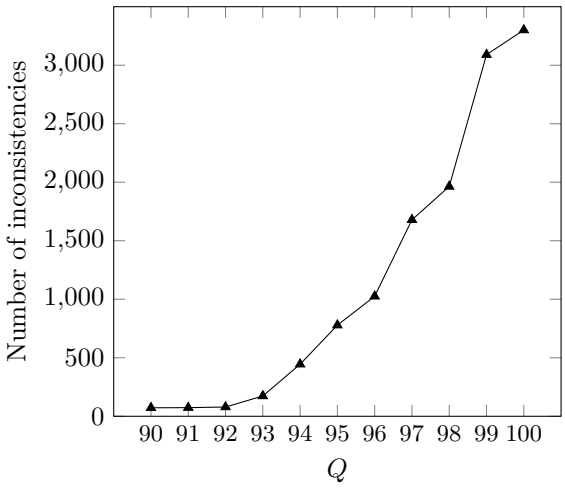

Figure 8.6.2: Average number of inconsistencies across 1000 randomly selected images from BOSSbase with $Q_1 = Q_2$.

Q	Detector	DC-SI	JUNI
75	SRNet	0.0585	0.0573
	e-SRNet	0.5000	0.5000
90	SRNet	0.0947	0.0983
	e-SRNet	0.2041	0.5000
95	SRNet	0.1856	0.1704
	e-SRNet	0.0000	0.5000

Table 8.6: Detection error P_{E} with SRNet and e-SRNet of J-UNIWARD in SC images and SI-UNIWARD in DC images at 0.4 bpnzac and $Q_1 = Q_2$. BOSSbase+BOWS2 dataset.

JPEG image to the spatial domain $\mathbf{y}^{(2)}$ and round to integers $\mathbf{x}^{(2)} = [\mathbf{y}^{(2)}]$. We then compress $\mathbf{x}^{(2)}$ again with the same quality settings to obtain the DCT coefficients after the third compression $\tilde{\mathbf{c}}^{(3)} = \text{DCT}(\mathbf{x}^{(2)}) \oslash \mathbf{q}$, where $\mathbf{q}$ is the quantization table used in all compression steps. A simple estimate of the rounding error can be computed as $\hat{\mathbf{e}} = \tilde{\mathbf{c}}^{(3)} - [\tilde{\mathbf{c}}^{(3)}]$. This estimate $\hat{\mathbf{e}}$ is strongly correlated with the original $\mathbf{e}$. This is illustrated in Figure 8.6.3, which displays the mean square error (MSE) between the DCT rounding error and its estimate $\text{MSE}(\mathbf{e}, \hat{\mathbf{e}}) = \frac{1}{n}\sum_{i=1}^{n}(e_i - \hat{e}_i)^2$. The estimate is computed from cover images and SI-UNIWARD images embedded with 0.1 and 0.4 bpnzac. With increasing payload (increasing number of inconsistencies), the estimate of the errors is getting worse across all qualities, which confirms our insight. For a smaller payload, we have a better estimate of the side-information. To verify that the estimate $\hat{\mathbf{e}}$ can be used for estimating the selection channel, we include in Figure 8.6.4 the correlation between $\hat{\mathbf{e}}$ and the difference $\beta^+ - \beta^-$, where β^+, β^- are the probabilities of changing the coefficients by $+1$ and -1, respectively.

This should be thought of more as a proof of concept because the e-SRNet most likely does not compress the image for the third time as it might compute the estimate of the rounding errors in some other, perhaps better way. It turns out that a similar estimate can be achieved by compressing the spatial rounding error $\mathbf{u} = \mathbf{x}^{(2)} - \mathbf{y}^{(2)}$, which is what the e-SRNet is trained on, and computing the rounding error in the DCT domain.

Using $Q_1 = Q_2$ for qualities below 93 will not be beneficial because the rounding errors $\mathbf{e}$ follow the distribution (8.4.5), which for $Q_1 = Q_2$ can be simplified as

$$e_{kl} \sim \mathcal{N}_F\left(0, \frac{1}{12(q_{kl}^{(2)})^2}\right). \tag{8.6.1}$$

It should be clear that for large quantization steps the errors will be clustered very closely around zero, thus having a negligible effect on the embedding. Moreover, as already mentioned above, for lower qualities there are not many inconsistencies, which is also due to (8.6.1). Therefore, the image is virtually identical to its single compressed version and there is not much side-information available. All these observations would suggest that the steganographic security would be very close to the non-informed J-UNIWARD on SC images. This is indeed verified in Table 8.6 showing that the SRNet on SI-UNIWARD in DC images with $Q_1 = Q_2$ has almost the same performance as on J-UNIWARD in SC images. The only difference is for quality 95, where the side-informed version seems to be slightly more secure thanks to the side-information generated in modes with small quantization steps (8.6.1). However, at this high quality the RJCA is already kicking in for the DC images, while not yet for SC images [13, 21], making steganography in DC images highly detectable.

8.7 Conclusions

In this paper, we pursued an idea of improving empirical steganographic security by embedding into a recompressed JPEG image instead of the original single compressed image. This idea of generating

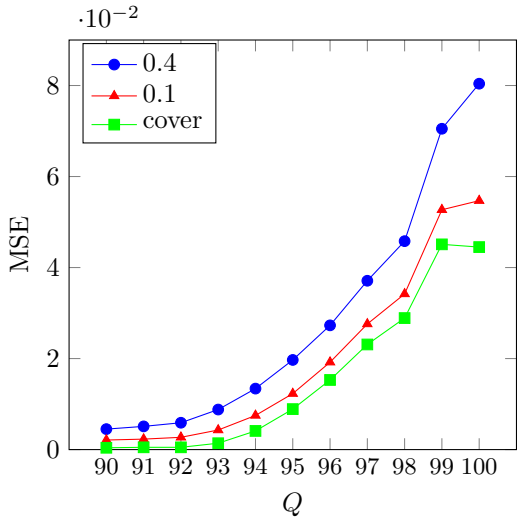

Figure 8.6.3: Average MSE between $\mathbf{e}$ and $\hat{\mathbf{e}}$ across 300 randomly selected images with $Q_1 = Q_2$. The estimate $\hat{\mathbf{e}}$ is computed from cover images and SI-UNIWARD at 0.1 and 0.4 bpnzac with $Q_1 = Q_2$. BOSSbase+BOWS2 dataset.

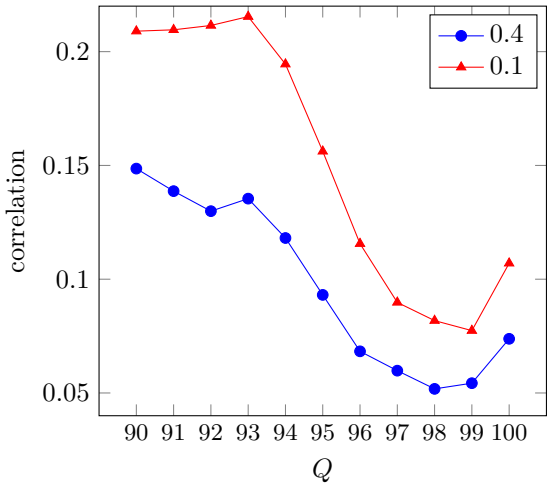

Figure 8.6.4: Correlation between $\hat{\mathbf{e}}$ and $\beta^+ - \beta^-$ across 300 randomly selected images for SI-UNIWARD at 0.1 and 0.4 bpnzac with $Q_1 = Q_2$. BOSSbase+BOWS2 dataset.

steganographic side-information by recompressing the JPEG cover image was first explored in the so-called Perturbed Quantization steganography 17 years ago. Surprisingly, when simply adopting modern coding coupled with cost modulation typically used in side-informed embedding, the security of the resulting embedding is extremely poor. This tells us that the side-information generated by recompression needs to be treated differently.

By quantifying the effect of embedding on the distribution of DCT coefficients from specific DCT modes using the steganographic Fisher information, we learned that modes that do not contain any contributing DCT coefficients (coefficients with rounding errors close in absolute value to 1/2 during recompression) exhibit artifacts in their distribution after embedding, which brings the security down. This was remedied by constraining the embedding only to contributing modes. Besides dramatically improving the security, this choice also allowed embedding larger, and thus more practical, payloads than embedding only into contributing DCT coefficients akin to the original PQ. To demonstrate the usefulness of the proposed technique, the empirical security was compared with embedding into the single compressed cover image while fixing the absolute payload in bits.

The method was also adapted for color images with the CCM payload-spreading strategy. To achieve a good security, however, the stabilizing constant of the J-UNIWARD algorithm had to be modified for the chrominance channels due to their different dynamic range.

Finally, we show that generating the side-information by recompressing with the same quantization table makes the embedding algorithm much more detectable because in such cases the side-information can be reliably estimated. This also leads to a bizarre situation for qualities below 93 when the detection power increases with smaller payloads.

Chapter 9

Conclusion

TODO

Bibliography

[1] H. Abdulrahman, M. Chaumont, P. Montesinos, and B. Magnier. Color image steganalysis using correlations between RGB channels. In *Proceedings 10th International Conference on Availability, Reliability and Security (ARES), 4th International Workshop on Cyber Crime (IWCC)*, pages 448–454, Toulouse, France, August 24–28 2015.

[2] H. Abdulrahman, M. Chaumont, P. Montesinos, and B. Magnier. Color image steganalysis using RGB channel geometric transformation measures. *Wiley Journal on Security and Communication Networks*, February 2016.

[3] S. Agarwal and H. Farid. Photo forensics from rounding artifacts. ACM Press, 2020.

[4] P. Bas, T. Filler, and T. Pevný. Break our steganographic system – the ins and outs of organizing BOSS. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, volume 6958 of Lecture Notes in Computer Science, pages 59–70, Prague, Czech Republic, May 18–20, 2011.

[5] P. Bas and T. Furon. BOWS-2. http://bows2.ec-lille.fr, July 2007.

[6] R. Böhme. Weighted stego-image steganalysis for JPEG covers. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lecture Notes in Computer Science, pages 178–194, Santa Barbara, CA, June 19–21, 2007. Springer-Verlag, New York.

[7] R. Böhme. *Advanced Statistical Steganalysis*. Springer-Verlag, Berlin Heidelberg, 2010.

[8] M. Boroumand, M. Chen, and J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, May 2019.

[9] M. Boroumand and J. Fridrich. Synchronizing embedding changes in side-informed steganography. In *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2020*, San Francisco, CA, January 26–30 2020.

[10] S. Bozinovski and A. Fulgosi. The influence of pattern similarity and transfer learning upon training of a base perceptron b2. In *Proceedings of Symposium Informatica 3-121-5*, 1976.

[11] J. Butora and J. Fridrich. Detection of diversified stego sources using CNNs. In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2019*, San Francisco, CA, January 14–17, 2019.

[12] J. Butora and J. Fridrich. Effect of jpeg quality on steganographic security. In R. Cogranne and L. Verdoliva, editors, *The 7th ACM Workshop on Information Hiding and Multimedia Security*, Paris, France, July 3–5, 2019. ACM Press.

[13] J. Butora and J. Fridrich. Reverse JPEG compatibility attack. *IEEE Transactions on Information Forensics and Security*, 15:1444–1454, 2020.

[14] J. Butora and J. Fridrich. Steganography and its detection in JPEG images obtained with the "trunc" quantizer. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, May 4–8, 2020.

[15] J. Butora and J. Fridrich. Extending the reverse JPEG compatibility attack to double compressed images. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Canada, June 6–11, 2021.

[16] C. Cachin. An information-theoretic model for steganography. *Information and Computation*, 192(1):41–56, July 2004.

[17] Marc Chaumont. Deep learning in steganography and steganalysis from 2015 to 2018. 2019.

[18] M. Chen, M. Boroumand, and J. Fridrich. Reference channels for steganalysis of images with convolutional neural networks. Lecture Notes in Computer Science, 2019. Under review.

[19] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich. JPEG-phase-aware convolutional neural network for steganalysis of JPEG images. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017.

[20] K. Chubachi. An ensemble model using CNNs on different domains for ALASKA2 image steganalysis. In *IEEE International Workshop on Information Forensics and Security*, New York, NY, December 6–11, 2020.

[21] R. Cogranne. Selection-channel-aware reverse JPEG compatibility for highly reliable steganalysis of JPEG images. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, pages 2772–2776, Barcelona, Spain, May 4–8, 2020.

[22] R. Cogranne, Q. Giboulot, and P. Bas. The ALASKA steganalysis challenge: A first step towards steganalysis "Into the wild". In R. Cogranne and L. Verdoliva, editors, *The 7th ACM Workshop on Information Hiding and Multimedia Security*, Paris, France, July 3–5, 2019. ACM Press.

[23] R. Cogranne, Q. Giboulot, and P. Bas. ALASKA–2: Challenging academic research on steganalysis with realistic images. In *IEEE International Workshop on Information Forensics and Security*, New York, NY, December 6–11, 2020.

[24] R. Cogranne, Q. Giboulot, and P. Bas. Steganography by minimizing statistical detectability: The cases of jpeg and color images. ACM Press, 2020.

[25] R. Cogranne and F. Retraint. An asymptotically uniformly most powerful test for LSB Matching detection. *IEEE Transactions on Information Forensics and Security*, 8(3):464–476, 2013.

[26] R. Cogranne, V. Sedighi, T. Pevný, and J. Fridrich. Is ensemble classifier needed for steganalysis in high-dimensional feature spaces? In *IEEE International Workshop on Information Forensics and Security*, Rome, Italy, November 16–19, 2015.

[27] R. Cogranne, C. Zitzmann, L. Fillatre, F. Retraint, I. Nikiforov, and P. Cornu. A cover image model for reliable steganalysis. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 178–192, Prague, Czech Republic, May 18–20, 2011.

[28] R. Cogranne, C. Zitzmann, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu. Statistical detection of LSB Matching using hypothesis testing theory. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 46–62, Berkeley, California, May 15–18, 2012.

[29] J. L. Davidson and P. Parajape. Double-compressed JPEG detection in a steganalysis system. In *Annual ADFSL Conference on Digital Forensics, Security, and Law*, May 30, 2012.

[30] T. Denemark, M. Boroumand, and J. Fridrich. Steganalysis features for content-adaptive JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 11(8):1736–1746, August 2016.

[31] T. Denemark and J. Fridrich. Side-informed steganography with additive distortion. In *IEEE International Workshop on Information Forensics and Security*, Rome, Italy, November 16–19 2015.

[32] T. Denemark and J. Fridrich. Improving selection-channel-aware steganalysis features. In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2016*, San Francisco, CA, February 14–18, 2016.

[33] T. Denemark and J. Fridrich. Model based steganography with precover. In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2017*, San Francisco, CA, January 29–February 1, 2017.

[34] T. Denemark, J. Fridrich, and V. Holub. Further study on the security of S-UNIWARD. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2014*, volume 9028, pages 05 1–13, San Francisco, CA, February 3–5, 2014.

[35] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich. Selection-channel-aware rich model for steganalysis of digital images. In *IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, December 3–5, 2014.

[36] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pages 248–255, June 20–25, 2009.

[37] S. Dumitrescu and X. Wu. LSB steganalysis based on higher-order statistics. In A. M. Eskicioglu, J. Fridrich, and J. Dittmann, editors, *Proceedings of the 7th ACM Multimedia & Security Workshop*, pages 25–32, New York, NY, August 1–2, 2005.

[38] S. Dumitrescu, X. Wu, and N. D. Memon. On steganalysis of random LSB embedding in continuous-tone images. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2002*, pages 324–339, Rochester, NY, September 22–25, 2002.

[39] L. Fillatre. Adaptive steganalysis of least significant bit replacement in grayscale images. *IEEE Transactions on Signal Processing*, 60(2):556–569, 2011.

[40] T. Filler and J. Fridrich. Fisher information determines capacity of ϵ-secure steganography. In S. Katzenbeisser and A.-R. Sadeghi, editors, *Information Hiding, 11th International Conference*, volume 5806 of Lecture Notes in Computer Science, pages 31–47, Darmstadt, Germany, June 7–10, 2009. Springer-Verlag, New York.

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

[42] T. Filler, A. D. Ker, and J. Fridrich. The Square Root Law of steganographic capacity for Markov covers. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Media Forensics and Security*, volume 7254, pages 08 1–11, San Jose, CA, January 18–21, 2009.

[43] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, 2009.

[44] J. Fridrich. On the role of side-information in steganography in empirical covers. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2013*, volume 8665, pages 1–11, San Francisco, CA, February 5–7, 2013.

[45] J. Fridrich and R. Du. Secure steganographic methods for palette images. In A. Pfitzmann, editor, *Information Hiding, 3rd International Workshop*, volume 1768 of Lecture Notes in Computer Science, pages 47–60, Dresden, Germany, September 29–October 1, 1999. Springer-Verlag, New York.

[46] J. Fridrich and M. Goljan. On estimation of secret message length in LSB steganography in spatial domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 23–34, San Jose, CA, January 19–22, 2004.

[47] J. Fridrich, M. Goljan, and R. Du. Detecting LSB steganography in color and gray-scale images. *IEEE Multimedia, Special Issue on Security*, 8(4):22–28, October–December 2001.

[48] J. Fridrich, M. Goljan, and R. Du. Steganalysis based on JPEG compatibility. In A. G. Tescher, editor, *Special Session on Theoretical and Practical Issues in Digital Watermarking and Data Hiding, SPIE Multimedia Systems and Applications IV*, volume 4518, pages 275–280, Denver, CO, August 20–24, 2001.

[49] J. Fridrich, M. Goljan, and D. Hogea. Attacking the OutGuess. In *Proceedings of the ACM, Special Session on Multimedia Security and Watermarking*, Juan-les-Pins, France, December 6, 2002.

[50] J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography using wet paper codes. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 6th ACM Multimedia & Security Workshop*, pages 4–15, Magdeburg, Germany, September 20–21, 2004.

[51] J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography. *ACM Multimedia System Journal*, 11(2):98–107, 2005.

[52] J. Fridrich, M. Goljan, D. Soukal, and P. Lisoněk. Writing on wet paper. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 328–340, San Jose, CA, January 16–20, 2005.

[53] J. Fridrich, M. Goljan, D. Soukal, and P. Lisoněk. Writing on wet paper. In T. Kalker and P. Moulin, editors, *IEEE Transactions on Signal Processing, Special Issue on Media Security*, volume 53, pages 3923–3935, October 2005. (journal version).

[54] J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2011.

[55] J. Fridrich and J. Kodovský. Steganalysis of LSB replacement using parity-aware features. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 31–45, Berkeley, California, May 15–18, 2012.

[56] J. Fridrich and J. Kodovský. Multivariate Gaussian model for designing additive distortion for steganography. In *Proc. IEEE ICASSP*, Vancouver, BC, May 26–31, 2013.

[57] J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.

[58] C. Fuji-Tsang and J. Fridrich. Steganalyzing images of arbitrary size with CNNs. In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2018*, San Francisco, CA, January 29–February 1, 2018.

[59] Q. Giboulot, R. Cogranne, and P. Bas. JPEG steganography with side information from the processing pipeline. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, pages 2767–2771, Barcelona, Spain, May 4–8, 2020.

[60] Q. Giboulot, R. Cogranne, and P. Bas. Synchronization Minimizing Statistical Detectability for Side-Informed JPEG Steganography. In *IEEE International Workshop on Information Forensics and Security*, New York, NY, December 6–11, 2020.

[61] Q. Giboulot, R. Cogranne, D. Borghys, and P. Bas. Effects and Solutions of Cover-Source Mismatch in Image Steganalysis. *Signal Processing: Image Communication*, August 2020.

[62] M. Goljan, R. Cogranne, and J. Fridrich. Rich model for steganalysis of color images. In *Sixth IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, December 3–5, 2014.

[63] M. Goljan and J. Fridrich. Cfa-aware features for steganalysis of color images. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[64] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. 2014.

[65] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. 2015.

[66] L. Guo, J. Ni, and Y.-Q. Shi. An efficient JPEG steganographic scheme using uniform embedding. In *Fourth IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5, 2012.

[67] L. Guo, J. Ni, and Y. Q. Shi. Uniform embedding for efficient JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 9(5):814–825, May 2014.

[68] S. Hetzl and P. Mutzel. A graph-theoretic approach to steganography. In J. Dittmann, S. Katzenbeisser, and A. Uhl, editors, *Communications and Multimedia Security, 9th IFIP TC-6 TC-11 International Conference, CMS 2005*, volume 3677 of Lecture Notes in Computer Science, pages 119–128, Salzburg, Austria, September 19–21, 2005.

[69] V. Holub. *Content Adaptive Steganography – Design and Detection.* PhD thesis, Binghamton University, May 2014.

[70] V. Holub and J. Fridrich. Designing steganographic distortion using directional filters. In *Fourth IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5, 2012.

[71] V. Holub and J. Fridrich. Challenging the doctrines of JPEG steganography. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2014*, volume 9028, pages 02–1–02–7, San Francisco, CA, February 3–5, 2014.

[72] V. Holub and J. Fridrich. Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Transactions on Information Forensics and Security*, 10(2):219–228, February 2015.

[73] V. Holub and J. Fridrich. Phase-aware projection model for steganalysis of JPEG images. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[74] V. Holub, J. Fridrich, and T. Denemark. Universal distortion design for steganography in an arbitrary domain. *EURASIP Journal on Information Security, Special Issue on Revised Selected Papers of the 1st ACM IH and MMS Workshop*, 2014:1, 2014.

[75] X. Hu, J. Ni, W. Su, and J. Huang. Model-based image steganography using asymmetric embedding scheme. *Journal of Electronic Imaging*, 27(4):1 – 7, 2018.

[76] F. Huang, J. Huang, and Y.-Q. Shi. New channel selection rule for JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 7(4):1181–1191, August 2012.

[77] F. Huang, W. Luo, J. Huang, and Y.-Q. Shi. Distortion function designing for JPEG steganography with uncompressed side-image. In W. Puech, M. Chaumont, J. Dittmann, and P. Campisi, editors, *1st ACM IH&MMSec. Workshop*, Montpellier, France, June 17–19, 2013.

[78] A. D. Ker. Improved detection of LSB steganography in grayscale images. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of Lecture Notes in Computer Science, pages 97–115, Toronto, Canada, May 23–25, 2004. Springer-Verlag, Berlin.

[79] A. D. Ker. A general framework for structural analysis of LSB replacement. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Information Hiding, 7th International Workshop*, volume 3727 of Lecture Notes in Computer Science, pages 296–311, Barcelona, Spain, June 6–8, 2005. Springer-Verlag, Berlin.

[80] A. D. Ker. Resampling and the detection of LSB matching in color bitmaps. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 1–15, San Jose, CA, January 16–20, 2005.

[81] A. D. Ker. Steganalysis of LSB matching in grayscale images. *IEEE Signal Processing Letters*, 12(6):441–444, June 2005.

[82] A. D. Ker. A fusion of maximal likelihood and structural steganalysis. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of Lecture Notes in Computer Science, pages 204–219, Saint Malo, France, June 11–13, 2007. Springer-Verlag, Berlin.

[83] A. D. Ker. Optimally weighted least-squares steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 6 1–6 16, San Jose, CA, January 29–February 1, 2007.

[84] A. D. Ker. Locating steganographic payload via WS residuals. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 27–32, Oxford, UK, September 22–23, 2008.

[85] A. D. Ker. Estimating steganographic fisher information in real images. In S. Katzenbeisser and A.-R. Sadeghi, editors, *Information Hiding, 11th International Conference*, volume 5806 of Lecture Notes in Computer Science, pages 73–88, Darmstadt, Germany, June 7–10, 2009. Springer-Verlag, New York.

[86] A. D. Ker. On the relationship between embedding costs and steganographic capacity. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017. ACM Press.

[87] A. D. Ker. The square root law of steganography. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017. ACM Press.

[88] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In E. J. Delp, P. W. Wong, J. Dittmann, and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 5 1–17, San Jose, CA, January 27–31, 2008.

[89] A. D. Ker, T. Pevný, J. Kodovský, and J. Fridrich. The Square Root Law of steganographic capacity. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 107–116, Oxford, UK, September 22–23, 2008.

[90] A. D. Ker and T. Pevnyý. A mishmash of methods for mitigating the model mismatch mess. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2014*, volume 9028, pages 1601–1615, San Francisco, CA, February 3–5, 2014.

[91] Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Science, pages 314–327, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.

[92] J. KodovskÃœ, V. Sedighi, and J. Fridrich. Study of cover source mismatch in steganalysis and ways to mitigate its impact. In A. M. Alattar, N. D. Memon, and C. D. Heitzenrater, editors, *Media Watermarking, Security, and Forensics 2014*, volume 9028, pages 204 – 215. International Society for Optics and Photonics, SPIE, 2014.

[93] J. Kodovský and J. Fridrich. On completeness of feature spaces in blind steganalysis. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 123–132, Oxford, UK, September 22–23, 2008.

[94] J. Kodovský and J. Fridrich. Calibration revisited. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 63–74, Princeton, NJ, September 7–8, 2009.

[95] J. Kodovsky and J. Fridrich. JPEG-compatibility steganalysis using block-histogram of recompression artifacts. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 78–93, Berkeley, California, May 15–18, 2012.

[96] J. Kodovský and J. Fridrich. Steganalysis of JPEG images using rich models. In A. Alattar, N. D. Memon, and E. J. Delp, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2012*, volume 8303, pages 0A 1–13, San Francisco, CA, January 23–26, 2012.

[97] J. Kodovský, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444, April 2012.

[98] B. Li, M. Wang, and J. Huang. A new cost function for spatial image steganography. In *Proceedings IEEE, International Conference on Image Processing, ICIP*, Paris, France, October 27–30, 2014.

[99] B. Li, W. Wei, A. Ferreira, and S. Tan. ReST-Net: Diverse activation modules and parallel subnets-based CNN for spatial image steganalysis. *IEEE Signal Processing Letters*, 25(5):650–654, May 2018.

[100] X. Li, T. Zeng, and B. Yang. Detecting LSB matching by applying calibration technique for difference image. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 133–138, Oxford, UK, September 22–23, 2008.

[101] P. Lu, X. Luo, Q. Tang, and L. Shen. An improved sample pairs method for detection of LSB embedding. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of Lecture Notes in Computer Science, pages 116–127, Toronto, Canada, May 23–25, 2004. Springer-Verlag, Berlin.

[102] I. Lubenko and A. D. Ker. Steganalysis with mismatched covers: Do simple classifiers help. In J. Dittmann, S. Katzenbeisser, and S. Craver, editors, *Proc. 13th ACM Workshop on Multimedia and Security*, pages 11–18, Coventry, UK, September 6–7 2012.

[103] W. Luo, Y. Wang, and J. Huang. Security analysis on spatial ±1 steganography for JPEG decompressed images. *IEEE Signal Processing Letters*, 18(1):39–42, 2011.

[104] S. Meignen and H. Meignen. On the modeling of DCT and subband image data for compression. *IEEE Transactions on Image Processing*, 4(2):186–193, February 1995.

[105] T. Mingxing and V. L. Quoc. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, volume 97, pages 6105–6114, June 9–15, 2019.

[106] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015.

[107] S. Ozcan and A. F. Mustacoglu. Transfer learning effects on image steganalysis with pre-trained deep residual neural network model. In *IEEE International Conference on Big Data (Big Data)*, pages 2280–2287, December 10–13, 2018.

[108] W. Pennebaker and J. Mitchell. *JPEG: Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, 1993.

[109] T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. In R. Böhme and R. Safavi-Naini, editors, *Information Hiding, 12th International Conference*, volume 6387 of Lecture Notes in Computer Science, pages 161–177, Calgary, Canada, June 28–30, 2010. Springer-Verlag, New York.

[110] L. Pibre, P. JÃ©rÃŽme, D. Ienco, and M. Chaumont. Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source-mismatch. 2018.

[111] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, Washington, DC, August 13–17, 2001.

[112] Y. Qian, J. Dong, W. Wang, and T. Tan. Deep learning for steganalysis via convolutional neural networks. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[113] V. Sachnev, H. J. Kim, and R. Zhang. Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 131–140, Princeton, NJ, September 7–8, 2009.

[114] P. Sallee. Model-based methods for steganography and steganalysis. *International Journal of Image Graphics*, 5(1):167–190, 2005.

[115] V. Sedighi, R. Cogranne, and J. Fridrich. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 11(2):221–234, 2016.

[116] V. Sedighi, J. Fridrich, and R. Cogranne. Content-adaptive pentary steganography using the multivariate generalized Gaussian cover model. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[117] C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec.*, 4:142–163, 1959.

[118] G. J. Simmons. The prisoner's problem and the subliminal channel. In D. Chaum, editor, *Advances in Cryptology, CRYPTO '83*, pages 51–67, Santa Barbara, CA, August 22–24, 1983. New York: Plenum Press.

[119] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang. Steganalysis of adaptive JPEG steganography using 2D Gabor filters. In P. Comesana, J. Fridrich, and A. Alattar, editors, *3rd ACM IH&MMSec. Workshop*, Portland, Oregon, June 17–19, 2015.

[120] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. 2014.

[121] T. Taburet, L. Filstroff, P. Bas, and W. Sawaya. An empirical study of steganography and steganalysis of color images in the JPEG domain. In *International Workshop on Digital Forensics and Watermarking (IWDW)*, Jeju, South Korea, 2018.

[122] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang. CNN-based adversarial embedding for image steganography. *IEEE Transactions on Information Forensics and Security*, 14(8):2074–2087, 2019.

[123] W. Tang, H. Li, W. Luo, and J. Huang. Adaptive steganalysis based on embedding probabilities of pixels. *IEEE Transactions on Information Forensics and Security*, 11(4):734–745, April 2016.

[124] T. Thai, R. Cogranne, and F. Retraint. Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm. *Image Processing, IEEE Transactions on*, 23(5):1–14, May 2014.

[125] T. H. Thai, R. Cogranne, and F. Retraint. Optimal detection of OutGuess using an accurate model of DCT coefficients. In *Sixth IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, December 3–5, 2014.

[126] D. Upham. Steganographic algorithm JSteg. Software available at http://zooid.org/ paul/crypto/jsteg.

[127] C. Wang and J. Ni. An efficient JPEG steganographic scheme based on the block–entropy of DCT coefficents. In *Proc. of IEEE ICASSP*, Kyoto, Japan, March 25–30, 2012.

[128] 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, Pittsburgh, PA, April 25–27, 2001. Springer-Verlag, New York.

[129] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In A. Pfitzmann, editor, *Information Hiding, 3rd International Workshop*, volume 1768 of Lecture Notes in Computer Science, pages 61–75, Dresden, Germany, September 29–October 1, 1999. Springer-Verlag, New York.

[130] G. Xu. Deep convolutional neural network to detect J-UNIWARD. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017.

[131] G. Xu, H. Z. Wu, and Y. Q. Shi. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712, May 2016.

[132] Y. Yang, X. Kong, and C. Feng. Double-compressed JPEG images steganalysis with transferring feature. *Multimedia Tools and Applications*, 77, February 2018.

[133] J. Ye, J. Ni, and Y. Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, November 2017.

[134] M. Yedroudj, M. Chaumont, and F. Comby. How to augment a small learning set for improving the performances of a CNN-based steganalyzer? In A. Alattar and N. D. Memon, editors, *Proceedings IS&T, Electronic Imaging, Media Watermarking, Security, and Forensics 2018*, San Francisco, CA, January 29–February 1, 2018.

[135] M. Yedroudj, F. Comby, and M. Chaumont. Yedroudj-net: An efficient CNN for spatial steganalysis. In *IEEE ICASSP*, pages 2092–2096, Alberta, Canada, April 15–20, 2018.

[136] Y. Yousfi, J. Butora, J. Fridrich, and C. F. Tsang. Improving EfficientNet for JPEG steganalysis. In *The 9th ACM Workshop on Information Hiding and Multimedia Security*, Brussels, Belgium, June 21–25, 2021. Under review.

[137] Y. Yousfi, J. Butora, Q. Giboulot, and J. Fridrich. Breaking ALASKA: Color separation for steganalysis in JPEG domain. In R. Cogranne and L. Verdoliva, editors, *The 7th ACM Workshop on Information Hiding and Multimedia Security*, Paris, France, July 3–5, 2019. ACM Press.

[138] Y. Yousfi, J. Butora, E. Khvedchenya, and J. Fridrich. Imagenet pre-trained cnns for jpeg steganalysis. In *IEEE International Workshop on Information Forensics and Security*, New York, NY, December 6–11, 2020.

[139] Y. Yousfi and J. Fridrich. An intriguing struggle of cnns in jpeg steganalysis and the onehot solution. *IEEE Signal Processing Letters*, 27:830–834, 2020.

[140] J. Zeng, S. Tan, B. Li, and J. Huang. Large-scale JPEG image steganalysis using hybrid deep-learning framework. *IEEE Transactions on Information Forensics and Security*, 13(5):1200–1214, May 2018.

[141] "X. Zhang, X. Kong, P. Wang, and B. Wang. Cover-source mismatch in deep spatial steganalysis. In H. Wang X. Zhao, Y. Shi, H. J. Kim, and A. Piva, editors, *Digital Forensics and Watermarking*, pages 71–83. Springer International Publishing, 2020.

[142] Y. Zhou, W. W. Y. Ng, and Z. He. Effects of double jpeg compression on steganalysis. In *International Conference on Wavelet Analysis and Pattern Recognition*, pages 106–112, 2012.

[143] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2021.

[144] C. Zitzmann, R. Cogranne, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu. Statistical decision methods in hidden information detection. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 163–177, Prague, Czech Republic, May 18–20, 2011.