Randomização Progressiva para Esteganálise

Anderson de Rezende Rocha

Dissertação de Mestrado

Departamento de Teoria da Computação
Instituto de Computação
Universidade Estadual de Campinas

Randomização Progressiva para Esteganálise

Anderson de Rezende Rocha[1]

17 de fevereiro de 2006

Banca Examinadora:

- Prof. Dr. Siome Klein Goldenstein
  IC – Unicamp (Orientador)
- Prof. Dr. Ricardo Dahab
  IC – Unicamp
- Prof. Dr. Eduardo Antônio Barros da Silva
  DEL / EE-PEE / COPPE – UFRJ
- Prof. Dr. Julio Cesar López Hernández
  IC – Unicamp (Suplente)

[1]**Financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) de Março de 2004 a Agosto de 2004 e pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo número 04/02384-1 de Setembro de 2004 a Fevereiro de 2006.**

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA CENTRAL DA UNICAMP
Bibliotecário: Helena Joana Flipsen – CRB-8ª / 5283

R582r

Rocha, Anderson de Rezende.
Randomização progressiva para esteganálise / Anderson de Rezende Rocha. -- Campinas, SP : [s.n.], 2006.

Orientador: Siome Klein Goldenstein.
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Randomização progressiva. 2. Imagens digitais - Detecção de conteúdo escondido. 3. Esteganálise. I.Goldenstein, Siome Klein. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Tradução do título em inglês: Progressive randomization for steganalysis.

Palavras-chave em inglês (Keywords): Progressive randomization, Hidden content detection, Steganalysis.

Área de concentração: Visão Computacional.

Titulação: Mestre em Ciência da Computação.

Banca examinadora: Siome Klein Goldenstein, Ricardo Dahab, Eduardo Antônio Barros da Silva, Julio Cesar López Hernández.

Data da defesa: 17-02-2006.

Randomização Progressiva para Esteganálise

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Anderson de Rezende Rocha e aprovada pela Banca Examinadora.

Campinas, 17 de fevereiro de 2006.

Prof. Dr. Siome Klein Goldenstein
IC – Unicamp (Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 17 de fevereiro de 2006, pela Banca examinadora composta pelos Professores Doutores:

Prof. Dr. Eduardo Antônio Barros da Silva
DEL/EE-PEE/COPPE - UFRJ

Prof. Dr. Ricardo Dahab
IC - UNICAMP

Prof. Dr. Siome Klein Goldenstein
IC - UNICAMP

Resumo

Neste trabalho, nós descrevemos uma nova metodologia para detectar a presença de conteúdo digital escondido nos *bits menos significativos* (LSBs) de imagens. Nós introduzimos a técnica de Randomização Progressiva (PR), que captura os artefatos estatísticos inseridos durante um processo de mascaramento com aleatoriedade espacial. Nossa metodologia consiste na progressiva aplicação de transformações de mascaramento nos LSBs de uma imagem. Ao receber uma imagem I como entrada, o método cria n imagens, que apenas se diferenciam da imagem original no canal LSB. Cada estágio da Randomização Progressiva representa possíveis processos de mascaramento com mensagens de tamanhos diferentes e crescente entropia no canal LSB. Analisando esses estágios, nosso arcabouço de detecção faz a inferência sobre a presença ou não de uma mensagem escondida na imagem I. Nós validamos nossa metodologia em um banco de dados com 20.000 imagens reais. Nosso método utiliza apenas descritores estatísticos dos LSBs e já apresenta melhor qualidade de classificação que os métodos comparáveis descritos na literatura.

Abstract

In this work, we describe a new methodology to detect the presence of hidden digital content in the *Least Significant Bits* (LSBs) of images. We introduce the Progressive Randomization technique that captures statistical artifacts inserted during the hiding process. Our technique is a progressive application of LSB modifying transformations that receives an image as input, and produces n images that only differ in the LSB from the initial image. Each step of the progressive randomization approach represents a possible content-hiding scenario with increasing size, and increasing LSB entropy. Analyzing these steps, our detection framework infers whether or not the input image I contains a hidden message. We validate our method with 20,000 real, non-synthetic images. Our method only uses statistical descriptors of LSB occurrences and already performs better than comparable techniques in the literature.

Agradecimentos

Toda caminhada é cheia de percalços e dificuldades. O que seria de nós se, nesses momentos, não pudéssemos contar com nossos amigos e colegas?

Este trabalho é de especial importância para mim. Estou realizando um sonho. O sonho de ser mestre. No entanto, só consegui chegar até aqui porque obtive ajuda de muitas pessoas. Nesse sentido, gostaria de agradecer a algumas delas, mesmo correndo o risco de deixar algumas de fora. A estas, desculpo-me antecipadamente.

Primeiramente, agradeço às duas pessoas mais importantes de minha vida: minha mãe Lucília e minha namorada Aninha. Vocês são minha inspiração.

Gostaria de agradecer também aos professores Cleide Abreu e Heitor Cantarella bem como aos amigos Milton Ferreira de Moraes, Dr. Nivaldo Baldo, Flávio Gomes (BH) e Regina Célia (BH) que foram os primeiros a me ajudar a vir para Campinas. Esta é uma cidade muito cara, principalmente para alguém de origem humilde como eu. Nos momentos em que estive sem bolsa, a ajuda destas pessoas foi fundamental.

Quero agradecer ao meu orientador Siome que desde o primeiro momento acreditou no potencial de nossa pesquisa e, mesmo sendo um tema novo, aceitou me orientar nesta área. Suas dicas foram muito importantes para o meu crescimento não só como estudante mas também como pesquisador e cidadão. Estendo meus agradecimentos aos professores com quem tive aulas no Instituto de Computação bem como ao professor Alexandre Falcão e aos colegas Paulo Miranda e Felipe Bergo com quem desenvolvi alguns trabalhos. Agradeço também aos meus colegas de apartamento Luís Meira e Wilson Pavon. Obrigado a todos pela amizade.

Durante meu projeto precisei da ajuda de muitas pessoas, principalmente durante a montagem do banco de imagens. Afinal, não é fácil conseguir 20.000 imagens. Neste sentido, gostaria de agradecer aos professores Rodolfo Azevedo e Jorge Stolfi pelas contribuições, assim como ao colega Renato Chencarek. Estendo meus agradecimentos ao colega Luís Meira pelas dicas na revisão teórica.

Agradeço à Unicamp. Esta é uma universidade que apóia o estudante em todos os momentos. É bom saber que o Brasil possui lugares como esse. Ajuda-nos a crer que o país tem jeito, basta acreditarmos. Finalmente, agradeço à FAPESP pelo apoio financeiro.

Epígrafe

Não basta ensinar ao homem uma especialidade, porque se tornará assim uma máquina utilizável e não uma personalidade. É necessário que adquira um sentimento, um senso prático daquilo que vale a pena ser empreendido, daquilo que é belo, do que é moralmente correto.

(*Albert Einstein*)

Dedicatória

Dedico este trabalho à minha mãe **Lucília** por ter adiado a realização de muitos de seus sonhos em benefício da realização dos meus. À ela toda a felicidade do mundo. Dedico também à minha namorada **Aninha**, por me fazer a pessoa mais feliz do mundo. Aninha, você é um presente para mim. Todos os dias agradeço sua presença em minha vida.

Conteúdo

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

De artificio sine secreti latentis suspicione scribendi! [1]
(David Kahn)

A busca por novos meios eficientes e eficazes de proteção digital é um campo de pesquisa fundamentado nas mais variadas áreas da ciência. Este campo de pesquisa se divide em duas ramificações. De um lado, estão aqueles que buscam técnicas para se obter maior proteção digital. Do outro lado, estão aqueles que querem minar a proteção, isto é, querem ter acesso à informação.

Uma das áreas que têm recebido muita atenção recentemente é a **esteganografia**. Esta é a arte de mascarar informações e evitar a sua detecção. Esteganografia deriva do grego, onde *estegano* = "esconder, mascarar" e *grafia* = "escrita". Logo, esteganografia é a arte da escrita encoberta.

A esteganografia inclui um vasto conjunto de métodos para comunicações secretas desenvolvidos ao longo da história. Entre tais métodos estão: tintas "invisíveis", micropontos, arranjo de caracteres (*character arrangement*), assinaturas digitais, canais escondidos (*covert channels*), comunicações por espalhamento de espectro (*spread spectrum communications*) entre outras.

Aplicações de esteganografia incluem identificação de componentes dentro de um subconjunto de dados, legendagem (*captioning*), rastreamento de documentos e certificação digital (*time-stamping*) e demonstração de que um conteúdo original não foi alterado (*tamper-proofing*). Entretanto, há indícios recentes de que a esteganografia tem sido utilizada para divulgar imagens de pornografia infantil na *internet* [22, 36].

Desta forma, é importante desenvolvermos algoritmos para detectar a existência de mensagens escondidas. Neste contexto, aparece a **esteganálise digital**, que se refere

[1] O artifício da comunicação secreta sem levantar suspeitas.

ao conjunto de técnicas que são desenvolvidas para distinguir entre objetos que possuem conteúdo escondido (estego-objetos) daqueles que não o possuem (não-estego).

As imagens digitais de cenas naturais possuem comportamento estatístico característico. Com a correta análise estatística, nós podemos determinar se uma imagem foi ou não alterada tornando as manipulações matematicamente detectáveis [33]. Neste caso, o objetivo da esteganálise em imagens é coletar informações estatísticas suficientes a respeito da presença de mensagens escondidas e usá-las para classificar se uma dada imagem de entrada contém ou não algum conteúdo escondido.

Em geral, é suficiente detectarmos a presença do conteúdo escondido em uma imagem. Por exemplo, agências de combate ao crime podem criar registros de acesso de conteúdos escondidos para construir uma rede de suspeitos. Posteriormente, utilizando outras técnicas, tais como inspeção física de material apreendido, pode-se descobrir os conteúdos escondidos e apreender as partes culpadas [24].

A inserção/modificação dos *bits* menos significativos (LSBs) é considerada a mais difícil de detectar [53, 38].

Neste trabalho, nós introduzimos uma nova metodologia para a detecção de conteúdo escondido no canal LSB de imagens digitais. Nós introduzimos a técnica de Randomização Progressiva (PR), que captura os artefatos estatísticos inseridos durante um processo de mascaramento com aleatoriedade espacial. Nossa metodologia consiste na progressiva aplicação de transformações de mascaramento nos LSBs de uma imagem. Ao receber uma imagem I como entrada, o método cria n imagens, que apenas se diferenciam da imagem original no canal LSB. Cada estágio da Randomização Progressiva representa possíveis processos de mascaramento com mensagens de tamanhos diferentes e crescente entropia no canal LSB. Analisando esses estágios, nosso arcabouço de detecção faz a inferência sobre a presença ou não de uma mensagem escondida na imagem I.

Nós validamos nossa metodologia em um banco de dados com 20.000 imagens reais. Nosso método utiliza apenas descritores estatísticos dos LSBs e já apresenta melhor qualidade de classificação que os métodos comparáveis descritos na literatura [23, 54, 18, 17, 31, 13].

No Capítulo 2, apresentamos uma revisão bibliográfica sobre esteganografia e esteganálise, incluindo as principais abordagens atualmente utilizadas, terminologia geral e alguns aspectos históricos. Descrevemos também algumas técnicas de classificação e aprendizado de máquina importantes para o entendimento deste trabalho. No Capítulo 3, introduzimos nossa metodologia de detecção de mensagens escondidas em imagens digitais e, no Capítulo 4, descrevemos o treinamento e validação de nosso arcabouço de detecção. Finalmente, apresentamos as conclusões, trabalhos futuros e extensões de nosso trabalho no Capítulo 5.

Capítulo 2

Revisão bibliográfica

Research is the art of seeing what everyone else has seen,
and doing what no-one else has done.[1]
(Anônimo)

Neste capítulo, nós apresentamos o conjunto de termos normalmente utilizados no campo de mascaramento digital de informações. Fazemos uma revisão bibliográfica sobre esteganografia e esteganálise apresentando as principais abordagens atualmente utilizadas. Descrevemos também algumas técnicas de classificação e aprendizado de máquina dado que algumas técnicas relacionadas à esteganálise interpretam este problema como um problema de classificação.

2.1 Terminologia

Segundo o **modelo geral de ocultamento de dados** (*information hiding*), **esteganografia** é a arte de esconder informações como uma forma de evitar a sua detecção. Esteganografia deriva do grego, donde *estegano = esconder, mascarar* e *grafia = escrita*. Logo, esteganografia é a arte da escrita encoberta. Em contrapartida, **esteganálise** é a arte de detectar mensagens escondidas nos mais diversos meios.

Dado embutido (*embedded data*) é a mensagem que desejamos enviar de maneira secreta. Freqüentemente, este dado é escondido em uma mensagem inócua (sem maior importância) conhecida como **mensagem de cobertura** (*cover-message*). As mensagens de cobertura podem mudar de nome de acordo com o meio de cobertura utilizado. Deste modo, podemos definir uma imagem de cobertura (*cover-image*), áudio de cobertura (*cover-audio*) ou texto de cobertura (*cover-text*). Após o processo de inserção dos

[1] Pesquisa é a arte de ver o que todos já viram e fazer o que ninguém fez.

dados na mensagem de cobertura, obtemos o chamado **estego-objeto** (*stego-object*), uma mensagem inócua contendo secretamente uma mensagem de maior importância [38]. Denominamos **mascaramento**, ao ato de esconder uma mensagem em um determinado meio.

Podemos utilizar uma **estego-chave** (*stego-key*) para controlar o processo de ocultamento de forma a restringir a detecção e/ou recuperação dos dados do material embutido. A Figura 2.1 apresenta como podemos interpretar o processo. Um indivíduo escolhe o dado a ser escondido e, a partir de uma chave, esconde estes dados em uma imagem de cobertura previamente selecionada. O resultado é a estego-imagem a ser enviada. Um

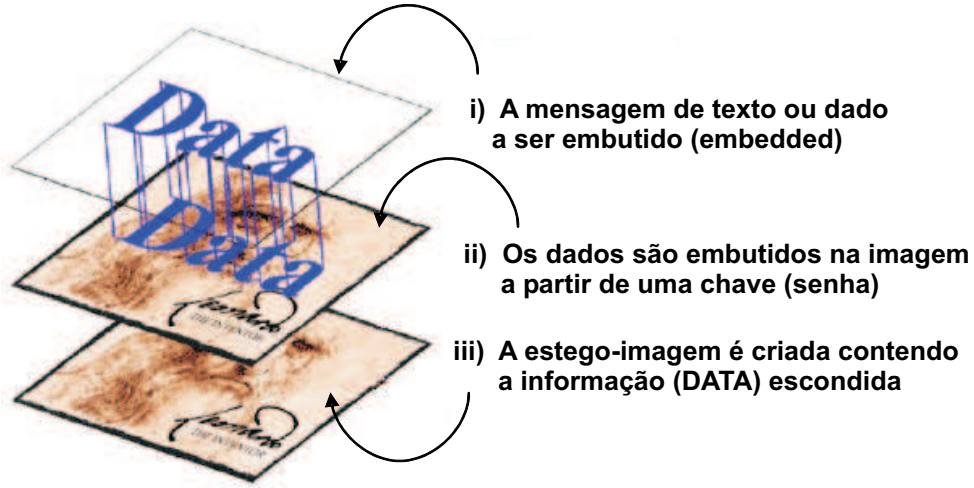


Figura 2.1: Exemplo de ocultamento de uma mensagem.

ataque com sucesso à esteganografia consiste em conseguir detectar a existência de uma mensagem escondida em algum meio observado. Outros sistemas, tais como marcação de direitos autorais (*watermarking*) têm requisitos adicionais de robustez contra possíveis ataques. Deste modo, um ataque bem-sucedido consiste em detectar e remover a marcação de direitos autorais [38].

O sistema de **seriação digital** (*fingerprinting*), também conhecido como etiquetas (*labels*), consiste em uma série de números embutidos no material a ser protegido. Isto permite identificar, por exemplo, se um cliente quebrou um acordo de propriedade intelectual.

Apresentamos, na Figura 2.2 [39], a grande-área de pesquisa conhecida como **ocultamento da informação** (*information hiding*). No segundo nível da hierarquia temos: canais secretos, esteganografia, anonimato e marcação de direitos autorais.

A criação de uma comunicação entre duas partes em que o meio é secreto e seguro constitui o que conhecemos por **canais secretos**. Um exemplo consiste nas conversações

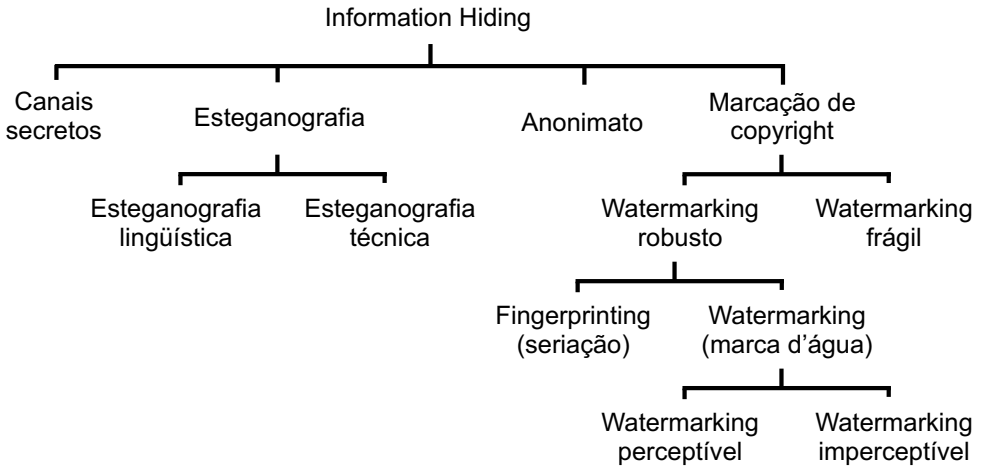


Figura 2.2: A hierarquia do ocultamento da informação.

militares em faixas de freqüências reservadas.

A arte da esteganografia constitui a segunda ramificação da hierarquia. Podemos dividi-la em lingüística e técnica. Quando a mensagem é fisicamente escondida, tal como a maioria dos exemplos que apresentamos na Seção 2.2, temos a chamada **esteganografia técnica**. Por outro lado, quando utilizamos propriedades lingüísticas para esconder a mensagem, (e.g. mensagens escondidas em *spams* e imagens), temos a chamada **esteganografia lingüística**.

Anonimato é um conjunto de técnicas para navegar na *internet*, por exemplo, sem ser localizado. Isto pode ser feito utilizando *sites* de desvio como o Anonymizer[2] e/ou *remailers* — *sites* capazes de enviar mensagens secretas não revelando seu remetente.

Marcação de direitos autorais (*copyright*) é a tentativa de manter ou provar a propriedade intelectual sobre algum tipo de mídia, seja esta eletrônica ou impressa. Neste sentido, **sistemas de marcação robustos** (*Watermarking robusto*) são aqueles que, mesmo após tentativas de remoção, permanecem intactos. Por outro lado, **sistemas de marcação frágeis** (*Watermarking frágil*) são aqueles em que qualquer modificação na mídia acarreta perda na marcação. Estes sistemas são úteis para impedir a cópia ilegal. Ao se copiar um material original, o resultado é um material não marcado e, por conseguinte, pirata. **Sistemas de marcação imperceptível** (*Watermarking imperceptível*) são aqueles em que as logomarcas dos autores, por exemplo, encontram-se no material, mas não são diretamente visíveis. Em contrapartida, **marcação visível** (*Watermarking visível*) é aquela em que o autor deseja mostrar sua autoria a todos que observarem a sua criação. Um exemplo desta última são imagens disponibilizadas na biblioteca do Vati-

[2] `http://www.anonymizer.com`

cano[3]. Nesta biblioteca, as imagens possuem um sistema de marcação digital visível [34], como pode ser observado na Figura 2.3.

Figura 2.3: Exemplo de marcação visível. Biblioteca do Vaticano.

2.2 Aspectos históricos

Durante toda a história, as pessoas têm tentado inúmeras formas de esconder informações dentro de outros meios, buscando, de alguma forma, mais privacidade para seus meios de comunicação [28, 37].

Um dos primeiros registros sobre esteganografia aparece em algumas descrições de Heródoto, o Pai da História, com vários casos sobre sua utilização. Um deles conta que um homem, de nome Harpagus, matou uma lebre e escondeu uma mensagem em suas entranhas. Em seguida, ele enviou a lebre através de seu mensageiro que se passou por um caçador [38].

Em outro caso, no século V (AC), um grego de nome Histaieus, a fim de encorajar Aristágoras de Mileto e seus compatriotas a começar uma revolta contra Medes e os persas, raspou a cabeça de um de seus escravos mais confiáveis e tatuou uma mensagem

[3] `http://bav.vatican.va`

em sua cabeça. Assim que os cabelos do escravo cresceram, o mesmo foi enviado à Mileto com instruções para que raspassem sua cabeça permitindo aos seus aliados receberem a mensagem [38].

Outra técnica interessante que aparece durante a História faz uso de inúmeras variações de tintas "invisíveis" (*invisible inks*). Tais tintas não são novidade e já apareciam em relatos de Plínio, o Velho, e Ovídio no século I (DC). Ovídio, em sua *Ars Amatoria*[4], propusera o uso do leite para escrita de textos "invisíveis". Para decodificar a mensagem, o receptor deveria borrifar o papel com ferrugem ou carbono negro. Estas substâncias aderiam ao leite e a mensagem era revelada [28, 27].

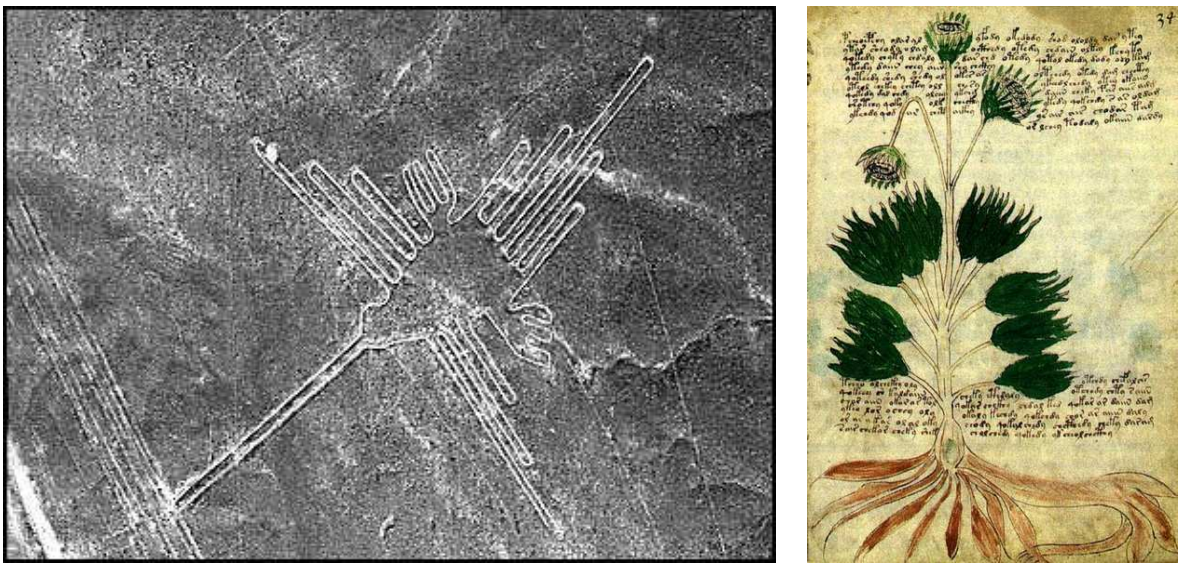
As primeiras tintas eram fluidos orgânicos que não exigiam nenhuma técnica especial para serem reveladas. Algumas vezes, bastava apenas aquecer o papel e a mensagem aparecia. Isto pode ser confirmado por meio de tintas baseadas em fluidos de suco de limão, por exemplo.

Na Segunda Guerra Mundial, com o aumento na qualidade das câmeras, lentes e filmes, tornou-se possível aos espiões nazistas, a criação de uma das formas mais interessantes e engenhosas de comunicação secreta. As mensagens nazistas eram fotografadas e, posteriormente, reduzidas ao tamanho de pontos finais (.). Assim, uma nova mensagem totalmente inocente era escrita contendo o filme ultra-reduzido como final das sentenças. A mensagem gerada era enviada sem levantar maiores suspeitas. Esta engenhosidade ficou conhecida como tecnologia do micro-ponto [48].

Em certas ocasiões, os emissores não possuem o interesse em esconder as mensagens. No entanto, se todos aqueles que são capazes de entendê-la deixarem de existir a mensagem torna-se, de alguma forma, escondida dado que não há mais quem a decifre. Neste sentido, podemos citar os "geoglifos" do platô de Nazca no Peru (Figura 2.4(a)), decifrados recentemente a partir de uma vista aérea [26], e o manuscrito de Voynich (Figura 2.4(b)), escrito em um alfabeto desconhecido por um autor anônimo há cerca de 600 anos e ainda não decifrado [45].

Atualmente, a esteganografia não foi esquecida. Ela foi modificada em sinal de acompanhamento aos novos tempos. Na era da informação, não faz mais sentido esconder mensagens dentro de lebres, inserir micro-pontos em uma revista ou mesmo utilizar tintas "invisíveis". Qualquer meio de esteganografia na atualidade, inevitavelmente, deve utilizar meios contemporâneos de tecnologia. Embora, em alguns casos, estes meios sejam apenas aperfeiçoamentos de técnicas clássicas.

[4] *Arte do amor.*



(a) (b)

Figura 2.4: (a) "Geoglifo" em forma de colibri, platô de Nazca, Peru. © Kiva Communications S.A. (b) Manuscrito de Voynich, seção "Herbal".

2.3 O estado da arte da esteganografia

As abordagens mais comuns de inserção de mensagens em imagens incluem técnicas de:

- inserção no *bit* menos significativo;
- técnicas de filtragem e mascaramento;
- algoritmos e transformações.

Cada uma destas técnicas pode ser aplicada a imagens, com graus variados de sucesso. O método de inserção no *bit* menos significativo é provavelmente uma das melhores técnicas de esteganografia em imagem [38, 53].

2.3.1 Inserção no bit menos significativo

Técnicas baseadas em LSB (*Least Significant Bit*) podem ser aplicadas a cada *byte* de uma imagem de 32-*bits*. Estas imagens possuem seus *pixels* codificados em quatro *bytes*. Um para o canal alfa (*alpha transparency*), outro para o canal vermelho (*red*), outro para o canal verde (*green*) e outro para o canal azul (*blue*). Seguramente, podemos selecionar um *bit* (o menos significativo) em cada *byte* do *pixel* para representar o *bit* a ser escondido

sem causar alterações perceptíveis na imagem. Estas técnicas constituem a forma de mascaramento em imagens mais difícil de ser detectada [40, 38, 53].

Acompanhe o exemplo da Figura 2.5. Suponha que desejemos esconder os *bits* **1110** dentro da área selecionada. Neste exemplo, sem perda de generalidade, utilizamos uma imagem em tons de cinza. Desta forma, temos um *bit* disponível para o mascaramento em cada *pixel* da imagem. Como queremos esconder quatro *bits*, precisamos selecionar quatro *pixels*. Para proceder o mascaramento, basta atribuirmos os *bits* selecionados de acordo com os *bits* que queremos esconder. Se o *bit* a ser escondido é 1 (0) então atribuímos o LSB do *pixel* selecionado para 1 (0). O principal objetivo de nossa pesquisa é a análise e detecção de mensagens escondidas nos *bits* menos significativos (LSBs) de uma imagem digital.

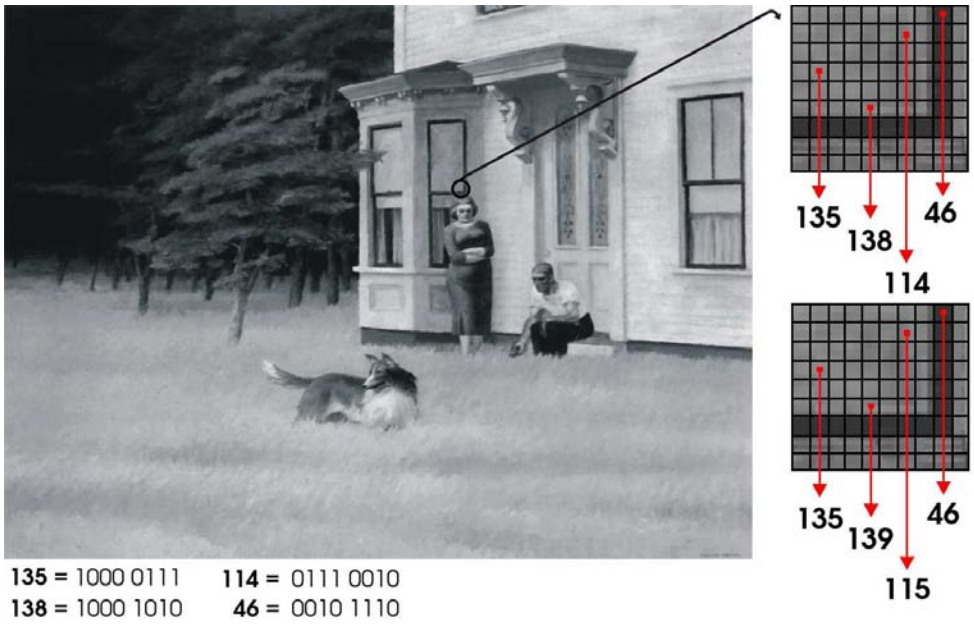


Figura 2.5: Um exemplo de mascaramento LSB para os *bits* 1110.

2.3.2 Técnicas de filtragem e mascaramento

Técnicas de esteganografia baseadas em filtragem e mascaramento são mais robustas que a inserção LSB. Estas técnicas geram estego-imagens imunes à compressão e recorte. No entanto, são técnicas mais propensas à detecção [53]. Ao contrário da inserção no canal LSB, técnicas de filtragem e mascaramento trabalham com modificações nos *bits* mais significativos das imagens. As imagens de cobertura devem ser em tons de cinza porque estas técnicas não são eficazes em imagens coloridas [40]. Isto deve-se ao fato de que

modificações em *bits* mais significativos de imagens em cores geram muitos artefatos[5] tornando as informações mais propensas à detecção.

2.3.3 Algoritmos e transformações

Técnicas de esteganografia baseadas em algoritmos e transformações tomam como aliado o principal inimigo da inserção no canal LSB: a compressão. Para isso, utilizamos a transformada discreta do cosseno, transformada discreta de Fourier e transformada Z [20], entre outras. Configuram-se como as mais sofisticadas técnicas de mascaramento de informações conhecidas [40, 23] embora sofisticação nem sempre implique em maior robustez aos ataques de esteganálise.

De forma geral, estas técnicas baseadas em algoritmos e transformações aplicam uma determinada transformação em blocos 8×8 *pixels* na imagem. Em cada bloco, selecionamos os coeficientes que são redundantes ou de menor importância. Posteriormente, utilizamos estes coeficientes para atribuir a mensagem a ser escondida em um processo em que cada coeficiente é substituído por um valor pré-determinado para o *bit* 0 ou o *bit* 1 [40].

2.3.4 Aplicativos de esteganografia disponíveis

Aplicações de esteganografia estão disponíveis na *internet* podendo ser executadas em uma grande variedade de plataformas incluindo DOS, Windows, Mac OS, Unix e Linux.

Ezstego e *Stego Online*[6] são duas ferramentas desenvolvidas na linguagem de programação *Java* e limitadas a imagens indexadas de oito *bits* e em formato GIF [6]. Estas duas ferramentas foram desenvolvidas por Romana Machado.

Henry Hastur desenvolveu duas outras ferramentas: *Mandelsteg* e *Stealth*[7]. *Mandelsteg* gera imagens de fractais para esconder as mensagens. *Stealth* é um programa que recebe uma mensagem criptografada pelo PGP [55], retira seu cabeçalho e a esconde em um arquivo. A retirada do cabeçalho de identificação é importante pois geralmente este contém informações do tipo de método criptográfico utilizado. Duas outras ferramentas capazes de trabalhar em associação com a criptografia são *Ray Arachelian's White Noise Storm*[8] e o *S-Tools*[9].

Colin Maroney desenvolveu o *Hide and Seek* [10]. Esta ferramenta é capaz de mascarar uma lista de arquivos em uma imagem, mas não faz uso de *criptografia*. Niels Provos

[5] Padrões.
[6] `http://www.stego.com`
[7] `ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/`
[8] `ftp.csua.berkeley.edu/pub/cypherpunks/steganography/wns210.zip`
[9] `ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/s-tools4.zip`
[10] `ftp://csua.berkeley.edu/pub/cypherpunks/steganography/hdsk41b.zip`

desenvolveu o *Outguess*[11] capaz de esconder mensagens com relativa robustez em imagens armazenadas no formato JPEG [25]. Testes estatísticos de primeira ordem não são capazes de detectar mensagens mascaradas com este *software* [43, 53].

Finalmente, duas outras ferramentas de destaque são *Jpeg-Jsteg* [12] capaz de fazer o mascaramento de informações utilizando os *pixels* mais significativos de uma imagem armazenada no formato JPEG e o *Camaleão*[13] [44] que utiliza cifragem de blocos e permutações cíclicas de modo a conseguir mascarar o conteúdo nos *bits* menos significativos de uma imagem armazenada no formato PNG [52].

2.4 Técnicas de esteganálise

Nesta seção, apresentamos uma visão geral das abordagens de esteganálise, os principais ataques e as técnicas estatísticas de detecção existentes atualmente. Ao final, descrevemos dois testes de significância (Sec. 2.5) que são de fundamental importância neste trabalho.

2.4.1 Visão geral

Grande parte das técnicas de esteganografia possuem falhas ou inserem artefatos (padrões) detectáveis nos objetos de cobertura. Algumas vezes, basta um **agressor** (alguém interessado em descobrir indevidamente a mensagem) fazer um exame mais detalhado destes artefatos para descobrir que há mensagens escondidas. Outras vezes, o processo de mascaramento de informações é mais robusto e as tentativas de detectar ou mesmo recuperar ilicitamente as mensagens podem ser bastante difíceis. Denominamos **esteganálise** o campo das pesquisas relacionado às tentativas de descobrir mensagens secretas numa alusão à **criptoanálise**, o campo de pesquisa relacionado à quebra de códigos e cifras [53, 54, 43].

Atualmente, as pesquisas em esteganálise estão concentradas em simplesmente identificar a presença de mensagens escondidas ao invés de extraí-las. Recuperar os dados escondidos, no momento, está além das capacidades da maioria dos testes uma vez que muitos algoritmos de mascaramento utilizam geradores aleatórios criptográficos muito seguros para misturar a informação no processo de mascaramento. Na maioria das vezes, os *bits* são espalhados pelo objeto de cobertura. Desta forma, os melhores algoritmos de *esteganálise* podem não ser capazes de dizer onde está a informação, mas devem dizer que os dados estão presentes.

A identificação da existência de uma mensagem escondida é suficiente para um agressor. As mensagens são, muitas vezes, frágeis e um agressor pode, sem muita dificuldade,

[11] `http://www.outguess.org/`
[12] `ftp.funet.fi/pub/crypt/steganography`
[13] `http://andersonrocha.cjb.net`

destruir a mensagem mesmo sem tê-la recuperado. Em algumas situações, entidades legais podem utilizar a identificação das mensagens escondidas para a construção de grafos de suspeitos que estão trocando mensagens escondidas em outros meios digitais pela *internet*. A partir dos registros das atividades suspeitas, pode-se conseguir mandados de busca e apreensão e devidamente apreender as partes culpadas.

Todos estes ataques dependem da identificação de algumas características em um objeto de cobertura (como imagens, vídeos, sons) que foram alteradas pelo processo de mascaramento. Não há qualquer garantia de que um algoritmo esteganográfico possa resistir à esteganálise.

> *Pode-se desenvolver um software que seja capaz de enganar todos os computadores uma vez, ou mesmo pode-se enganar alguns computadores todas as vezes. No entanto, nunca se poderá desenvolver um software capaz de enganar todos os computadores todas as vezes.* (Anônimo)

2.4.2 Tipos de ataques

Existem diversas abordagens para detectarmos a presença de conteúdo escondido em imagens digitais. Podemos dividir essas abordagens em três classes principais: (1) ataques aurais, (2) estruturais e (3) estatísticos. Descrevemos a seguir cada um desses ataques.

Ataques aurais

Estes ataques consistem em retirar as partes significativas da imagem como um meio de facilitar aos olhos humanos a busca por anomalias na imagem. Um teste comum é mostrar os *bits* menos significativos da imagem. Câmeras, *scanners* e outros meios digitalizadores sempre deixam alguns artefatos nos *bits* menos significativos[14].

O cérebro do ser humano é capaz de descobrir as mais sutis diferenças. Esta é a razão pela qual muitas marcações de áudio (*audio watermarking*) de grandes gravadoras são frustradas devido aos ouvidos de músicos bem-treinados.

Ataques Estruturais

A estrutura do arquivo de dados algumas vezes muda assim que outra mensagem é inserida. Nesses casos, um sistema capaz de analisar padrões estruturais seria capaz de descobrir a mensagem escondida. Por exemplo, ao escondermos mensagens em imagens indexadas (baseadas em paletas de cores), pode ser necessário usarmos diferentes versões de

[14]Grande parte de câmeras digitais ou mesmo digitalizadores utilizam os LSBs das imagens para acrescentar seus próprios padrões. Isto elimina o ruído existente e acrescenta os padrões desejados pelos fabricantes [53, 54, 49].

paletas. Este tipo de atitude muda as características estruturais da imagem de cobertura, logo as chances de detecção da presença de uma mensagem escondida aumentam [54, 53].

Ataques estatísticos

Os padrões dos *pixels* e seus *bits* menos significativos freqüentemente revelam a existência de uma mensagem secreta nos perfis estatísticos [53, 41, 42, 43]. Os novos dados não têm os mesmos perfis esperados.

Muitos dos estudos de *Matemática e Estatística* têm por objetivo classificar se um dado fenômeno ocorre ao acaso. Cientistas usam estas ferramentas para determinar se suas teorias explicam bem tal fenômeno. Estas técnicas estatísticas também podem ser usadas para determinar se uma dada imagem e/ou som possui alguma mensagem escondida. Na maioria das vezes, os dados escondidos são mais aleatórios que os dados que foram substituídos no processo de mascaramento ou inserem artefatos que alteram as propriedades estatísticas inerentes do objeto de cobertura [41, 42, 53, 17].

A seguir, apresentamos um resumo das principais técnicas de esteganálise baseadas em ataques estatísticos existentes. Para mais detalhes consulte o Apêndice A.

1. **Esteganálise por teste do χ^2 (*Chi-Square Test Approach*)**. Inicialmente apresentada Andreas Westfeld e Niels Provos [42, 54], esta técnica até o momento era limitada à detecção de mensagens escondidas seqüencialmente em imagens digitais. Nossa metodologia de detecção estende essa técnica para detecção de mensagens aleatoriamente escondidas nas imagens.

2. **RS Analysis**. Apresentada por Jessica Fridrich [16], esta técnica consiste na análise das inter-relações entre os planos de cores presente nas imagens analisadas. A classificação é feita pontualmente, sem utilização de treinamento e é dependente do contexto da imagem analisada.

3. **Taxa de inversão da energia do gradiente** (*Gradient Energy Flipping Rate*). Esta técnica foi desenvolvida por Li Shi [47]. Consiste em analisar a variação da energia do gradiente, devido ao processo de mascaramento, dos planos de *bits* das imagens analisadas.

4. **Análise de estatísticas de alta ordem**. Apresentada por Hany Farid e Siwei Lyu [12, 31, 30, 13] constitui um dos mais poderosos arcabouços de detecção até o momento. Baseia-se na decomposição da imagem em filtros de quadratura em espelho (QMFs – *Quadrature Mirror Filters*) [50]. Esta decomposição divide a imagem no domínio da freqüência em múltiplas escalas e orientações. Esta abordagem

pratica a chamada **detecção cega** (*blind detection*), ou seja, faz a detecção independentemente do *software* utilizado no processo de mascaramento. Para atingir seu objetivo, utiliza aprendizado e classificação.

5. **Métricas de qualidade de imagens** (*Image Quality Metrics*). Métricas de qualidade de imagem são utilizadas, de forma geral, na avaliação de codificação de artefatos, predição de performance de algoritmos de *Visão Computacional*, perda de qualidade devido a inadequabilidade de algum sensor, entre outras aplicações. Nesta abordagem proposta por Ismail Avcibas [1, 2, 3], essas mesmas métricas são utilizadas para construir um discriminador de imagens de cobertura (sem conteúdo escondido) de estego-imagens (com conteúdo escondido) através da utilização de *regressão multi-variada*. A classificação é feita por um discriminante linear após um certo treinamento (estabilização dos coeficientes da regressão multi-variada).

6. **Métricas de tons contínuos e análise de pares de amostragem** (*Continuous Tone Metrics and Sample Pair Analysis*). Proposta por Sorina Dumitrescu [11, 10], esta abordagem consiste em analisar as relações de identidade estatística existentes sobre alguns conjuntos de *pixels* considerados. As identidades observadas são muito sensíveis ao mascaramento LSB e as mudanças nestas identidades podem indicar a presença de conteúdo escondido.

A metodologia de detecção que apresentamos neste trabalho utiliza dois testes estatísticos de significância χ^2 e o *teste de Ueli* (U_T) sobre a imagem sendo analisada e são apresentados na Seção 2.5.

2.5 Testes de significância

Em Estatística, um resultado é considerado significante se ele é improvável de acontecer ao acaso dado que uma hipótese nula (H_0) é verdadeira. O nível de significância é a máxima probabilidade de acidentalmente rejeitar H_0 [15]. A seguir, mostramos dois testes de significância que foram utilizados em nosso trabalho: χ^2 e U_T.

2.5.1 Teste do χ^2

Proposto por Karl Pearson por volta de 1900 [15], este teste consiste em comparar duas freqüências f^{obs} e f^{esp} de mesmo tamanho, elemento a elemento. A freqüência f^{obs} representa nosso conjunto de observação e f^{esp} nosso conjunto de freqüências esperadas ou conjunto de comparação. O procedimento consiste em somar o quadrado das diferenças,

elemento a elemento e dividir por f_i^{esp}

$$\chi^2 = \sum_{i=1}^{\nu+1} \frac{(f_i^{obs} - f_i^{esp})^2}{f_i^{esp}}, \tag{2.1}$$

onde ν denota o número de elementos analisados. Quando f_i^{obs} está muito distante de f_i^{esp} (o valor observado está longe do valor esperado), o termo correspondente a esta diferença na soma será grande. Grandes valores de χ^2 podem indicar que a freqüência observada não está bem descrita pela freqüência esperada. Desta forma, o teste do χ^2 é uma medida da distância entre a freqüência observada e a freqüência esperada.

2.5.2 Teste de Ueli (U_T)

Proposto por Ueli Maurer em 1992 [32], este teste configura-se como uma abordagem bastante eficaz para detectar quão boa é uma seqüência pseudo-aleatória de números. É apresentado como o teste universal (U_T) para seqüências pseudo-aleatórias [32].

Em termos gerais, o procedimento do cálculo de U_T consiste em dividir uma seqüência de *bits* S em blocos. Seja $B(S) = (b_1, b_2, \ldots, b_N)$ tal que a concatenação dos elementos de B é S. Seja $|b_i| = L$ para cada i e $|B(S)| = N$. Definimos U_T como uma função $U_T : B(S) \rightarrow \Re^+$ da seguinte forma

$$U_T(B(S)) = \frac{1}{K} \sum_{i=Q}^{Q+K} \ln A(b_i), \tag{2.2}$$

onde

$$A(b_i) = \begin{cases} i, & \text{se não existe um inteiro positivo} \\ & i' < i \text{ tal que } b_{i'} = b_i, \\ \min\limits_{\forall i'}\{i' : b_{i'} = b_i\}, & \text{caso contrário,} \end{cases} \tag{2.3}$$

K é o número de *bits* analisados (na prática $K = N$), e Q é um valor de deslocamento feito na seqüência $B(S)$. Como sugerido no artigo [32], utilizamos $Q = \frac{K}{10}$.

Em seu trabalho [32], Maurer apresenta valores máximos esperados para U_T em função de L. Colocamos alguns destes valores na Tabela 2.1.

Quando avaliamos uma seqüência de *bits* S com $L = 8$, estamos avaliando uma seqüência de números entre $[0, \ldots, 2^8 - 1]$. Caso U_T seja próximo de 7.1836656, temos uma condição altamente randômica nesta seqüência. Por outro lado, quanto menor o valor de U_T, mais padronizada e menos aleatória é a condição de S.

L	U_T^{max}
1	0.7326495
4	3.3112247
8	7.1836656
16	15.167379

Tabela 2.1: Valor máximos esperados para diferentes Ls.

2.6 Cenários de ataque

Nas abordagens de ataque à esteganografia, podemos ter a combinação de diversos cenários [24]:

- **Apenas estego (*stego-only*)**. Apenas o meio estego está disponível para análise.
- **Cobertura conhecida (*known-cover*)**. Temos disponíveis para análise tanto o meio estego quanto o meio original utilizado para cobertura.
- **Mensagem conhecida (*known-message*)**. Quando, de alguma forma, temos em nosso poder uma mensagem escondida conhecida e a utilizamos para comparação em busca de padrões nos meios de cobertura.
- **Meio estego escolhido (*chosen-stego*)**. Quando geramos um meio estego a partir de um algoritmo esteganográfico conhecido e uma mensagem conhecida.
- **Mensagem escolhida (*chosen-message*)**. Não conhecemos a mensagem, conhecemos apenas o meio estego e o algoritmo de codificação utilizado.

2.7 Classificadores e aprendizado

Nesta seção, definimos o que é aprendizado de máquina e apresentamos algumas métricas de qualidade que podemos utilizar em um sistema de aprendizado. Descrevemos também os classificadores Análise do Discriminante Linear (LDA – *Linear Discriminant Analysis*), *Árvores de Classificação* (CTREES) e *Support Vector Machines* (SVMs) que utilizamos em nosso projeto. Apresentamos também o *Bagging*, uma poderosa abordagem de classificação baseada na replicação de classificadores mais fracos.

Em Matemática e Estatística, um **classificador** é um mapeamento a partir de um espaço de características X para um conjunto discreto de rótulos (*labels*) Y.

Em Inteligência Artificial, um classificador é um tipo de motor de inferência que implementa estratégias eficientes para computar relações de classificação entre pares de conceitos ou para computar relações entre um conceito e um conjunto de instâncias [9].

2.7.1 Aprendizado

Aprendizado de máquina é uma área da Inteligência Artificial concentrada no desenvolvimento de técnicas que permitem que computadores sejam capazes de aprender com a experiência [35]. Alguns problemas que utilizam aprendizado de máquina são: reconhecimento de caracteres, reconhecimento da fala, predição de ataques cardíacos e detecção de fraudes em cartões de créditos [35, 19]. Na solução desses problemas, podemos ter classificadores fixos ou baseados em aprendizado, que, por sua vez, pode ser supervisionado ou não-supervisionado [19].

Aprendizado supervisionado

É uma abordagem de aprendizado de máquina em que procuramos estimar uma função f de classificação a partir de um conjunto de treinamento. O conjunto de treinamento consiste de pares de valores de entrada X, e sua saída desejada Y [19]. De forma geral, denotamos saídas quantitativas por Y e qualitativas por G (grupo). Valores observados no conjunto X são denotados por x_i, isto é, x_i é a i-ésima observação em X. O número de variáveis que constituem cada uma das entradas X é p. Desta forma, X é formado por N vetores de entrada e cada vetor de entrada é composto por p graus de liberdade (dimensões e/ou variáveis).

A saída da função f pode ser um valor contínuo (regressão), ou pode predizer a etiqueta (label) de um objeto de entrada (classificação). A tarefa do aprendizado é predizer o valor da função para qualquer objeto de entrada que seja válido após ter sido suficientemente treinado com um conjunto de exemplos, ou seja, após ter visto uma quantidade razoável de entradas e suas respectivas saídas esperadas. O resultado da predição será $\hat{Y}$ para saídas quantitativas ou $\hat{G}$ para saídas qualitativas.

Para atingir seu objetivo, o motor de inferência que está sendo submetido ao aprendizado precisa **generalizar**, isto é, a partir de seu conhecimento adquirido, ele tem que ser capaz de responder razoavelmente à situações (entradas) novas [19].

Um problema constante no aprendizado supervisionado é quando parar de treinar. De forma geral, observamos que ao fornecer uma quantidade muito grande de exemplos de treinamento ao motor de inferência este torna-se incapaz de generalizar de forma razoável. Por outro lado, ao fornecermos poucos exemplos, o motor de inferência (classificador) torna-se muito geral e incapaz de predizer uma saída correta para novas entradas [19, 29].

Aprendizado não-supervisionado

É uma abordagem de aprendizado de máquina em que o motor de inferência não possui, *a priori*, uma saída conhecida. Esta forma de aprendizado, na maioria das vezes, trata o seu conjunto de entrada como um conjunto de variáveis aleatórias. Um modelo de distribuição conjunta (*joint distribution model*) é então construído para a representação dos dados. Desta forma, o objetivo deste aprendizado é avaliar como os dados estão organizados e agrupados (*clusters*) [19].

2.7.2 Redução de dimensionalidade

Em uma análise de dados, podemos estar trabalhando com muitas variáveis para cada exemplo que estamos analisando, isto é, o número de dimensões p analisado pode ser muito alto. Como proposto por Bellman em 1961 [4], o número de problemas associados com a análise de dados multi-variada cresce com a dimensionalidade do problema. Na prática, isto implica que para uma amostra de um certo tamanho há um número máximo de características a partir do qual a performance de nosso classificador irá piorar ao invés de melhorar. Outro fator limitante é que quanto maior a dimensionalidade do problema, maior o número de exemplos necessários para representar o espaço de análise. Um espaço muito esparso pode não conter informações suficientes para o treinamento de nosso classificador.

Nesses casos, podemos utilizar técnicas de redução de dimensionalidade de modo a eliminar características de nosso problema que não estão efetivamente contribuindo para a classificação de forma geral.

Há duas abordagens para efetuarmos a redução:

- *extração de características*. Cria um novo subconjunto de características através de combinações das características existentes. Um mapeamento ótimo $y = f(x)$ é aquele que não implica em um aumento significativo no erro de classificação. Em geral, esta função de extração é uma função não-linear. Entretanto, não existe uma maneira sistemática de gerar funções não-lineares. Desta forma, estamos interessados em uma aproximação linear para resolver este problema. Neste trabalho, utilizamos a abordagem de redução da análise discriminante linear (LDA – *Linear Discriminant Analysis*);

- *seleção de características*. Escolhe um subconjunto de características consideradas mais relevantes. Não consideramos essa abordagem neste trabalho.

Após a redução de dimensionalidade, todo o procedimento de classificação é feito no espaço de dimensão reduzida.

2.7.3 Validação

Ao utilizarmos aprendizado supervisionado para resolvermos um determinado problema, temos três opções de validarmos nossos resultados:

1. **Holdout validation**. Nesta abordagem, construímos nosso conjunto de treinamento a partir de observações sem repetição feitas aleatoriamente em nosso conjunto de dados iniciais. As observações remanescentes constituem nosso conjunto de validação ou de teste.

2. **K-fold cross-validation**. Nesta abordagem, nós dividimos o nosso conjunto inicial de dados em k partições e escolhemos uma partição como nosso conjunto de validação e as $k - 1$ partições restantes constituem nosso conjunto de treinamento. Então, repetimos o processo k vezes, com cada uma das partições sendo utilizada uma vez como conjunto de validação. Finalmente, calculamos a média (μ) e desvio padrão (σ) das k validações. Em todos os experimentos de validação que reportamos neste trabalho, nós utilizamos a validação cruzada com $k = 10$.

3. **Leave-one-out validation**. Como o nome sugere, este tipo de validação consiste em selecionar um elemento do conjunto de dados e utilizá-lo como conjunto de validação. Os $n - 1$ elementos restantes constituem nosso conjunto de treinamento. Repetimos o processo n vezes com cada um dos elementos do conjunto de dados sendo utilizado uma vez como conjunto de validação. Finalmente, calculamos a média (μ) e desvio padrão (σ) das n validações.

2.7.4 Métricas de qualidade

Considere um problema de classificação de duas classes Y(-1 | +1). De forma geral, existem quatro métricas bastante populares para se avaliar a qualidade de um classificador [29]:

1. *FNR* (*False Negative Rate*). É a percentagem de elementos que deveriam ser classificados como da classe +1 e foram classificados como da classe -1.

2. *TPR* (*True Positive Rate*). É a percentagem de elementos que são da classe +1 e foram corretamente classificados.

3. *FPR* (*False Positive Rate*). É a percentagem de elementos que deveriam ser classificados como da classe -1 e foram classificados como da classe +1.

4. *TNR* (*True Negative Rate*). É a percentagem de elementos que são da classe -1 e foram corretamente classificados.

Um bom classificador procura minimizar FNR e FPR e maximizar TNR e TPR. Sabemos que $FNR = 1 - TPR$ e $FPR = 1 - TNR$. Podemos definir E, a exatidão ou qualidade (*accuracy*) do sistema, como $E = (TP + TN)/N$, onde TP e TN são os elementos corretamente classificados pertencentes às classes $+1$ e -1, respectivamente.

A seguir, apresentamos os classificadores LDA, SVM, CTREE e *Bagging*, que utilizamos em nosso trabalho.

2.7.5 LDA

Também conhecida como discriminante de Fisher, esta técnica consiste em selecionar os componentes que maximizam a diferença entre as classes enquanto minimizam a variabilidade intra-classe [14, 19].

Por simplicidade, considere um problema de duas classes Y(-1 = a | +1 = b). Seja X_a uma família de exemplares pertencentes à classe a e X_b uma família de exemplares pertencentes à classe b no conjunto de treinamento. Considere $|X_a| = N_a$ e $|X_b| = N_b$.

Supondo que ambas as classes seguem uma distribuição Gaussiana, podemos definir as médias intra-classe como

$$\mu_a = \frac{1}{N_a} \sum_{x_i \in X_a} x_i \quad \text{e} \quad \mu_b = \frac{1}{N_b} \sum_{x_j \in X_b} x_j. \tag{2.4}$$

Definimos a média entre as classes (a, b) como

$$\mu = \frac{1}{N_a + N_b} \left(\sum_{x \in X_a \cup X_b} x \right). \tag{2.5}$$

Podemos definir a matriz S_w de dispersão (*scatter*) intra-classe como

$$S_w = M_a M_a^T + M_b M_b^T, \tag{2.6}$$

onde a i-ésima coluna da matriz M_a contém a diferença $(x_i^a - \mu_a)$. O mesmo procedimento se aplica a M_b. A matriz S_{bet} de dispersão entre as classes é definida como

$$S_{bet} = N_a(\mu_a - \mu)(\mu_a - \mu)^T + N_b(\mu_b - \mu)(\mu_b - \mu)^T. \tag{2.7}$$

Para maximizarmos a diferença entre as classes e minimizarmos a variabilidade intra-classe em uma simples dimensão, é suficiente calcularmos o autovalor-autovetor generalizado $\vec{e}$ maximal de S_{bet} e S_w, isto é, $S_{bet}\vec{e} = \lambda S_w \vec{e}$. Com o autovetor generalizado maximal, nós podemos projetar as amostras em um subespaço linear e aplicar um limiar para efetuarmos a classificação.

2.7.6 Árvores de classificação (CTREES)

A abordagem de árvores de classificação que descrevemos aqui utiliza propriedades estatísticas da teoria da informação para proceder a classificação. Alguns algoritmos conhecidos que aplicam esta abordagem são o ID3 e suas evoluções como as abordagens C4.5 e C5.0 [35]. Embora não descritas aqui, existem outras abordagens não baseadas na teoria da informação como a abordagem CART baseada em regressão univariada [19].

A maioria dos algoritmos baseados em árvores de classificação empregam técnicas gulosas *top-down* para analisar o espaço de possíveis soluções. Nessa abordagem de classificação, em cada iteração, procuramos selecionar os atributos que são mais propícios a serem nós da árvore.

Para a seleção de atributos, aplicamos um teste estatístico para avaliarmos cada atributo (característica) de modo a analisarmos quão bem ele classifica, sozinho, os exemplos presentes no treinamento. Por simplicidade, considere um problema de duas classes Y(-1 = a | +1 = b). Caso tenhamos N exemplos no treinamento e cada exemplo possua p características, no primeiro passo do algoritmo selecionamos o atributo que melhor classifica os N exemplos presentes no treinamento segundo o critério estatístico adotado. O processo é repetido para cada nó folha descendente produzido até que (1) o número total de atributos seja avaliado ou (2) todos os exemplos no treinamento associado a este determinado nó folha pertençam à mesma classe (e.g. todos os exemplos pertençam à classe $Y = +1$).

Ao utilizarmos árvores de classificação, temos que considerar dois pontos importantes (1) a seleção do atributo a ser testado em cada nó e (2) a seleção de limiares para atributos não-discretos.

Seleção do atributo a ser testado em cada nó

Para selecionarmos o atributo que melhor classifica o nosso conjunto de treinamento, definimos uma propriedade estatística chamada *ganho de informação* (GI) que mede quão bem um dado atributo separa os exemplos de treinamento de acordo com a classificação esperada. Para definirmos GI, definimos inicialmente uma medida comumente utilizada em teoria da informação, conhecida como *entropia*, que caracteriza o grau de organização em um conjunto arbitrário de exemplos. Dada uma coleção de exemplos S, contendo exemplos pertencentes tanto à nossa classe a quanto à nossa classe b, a entropia de S é definida como

$$E(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}, \tag{2.8}$$

onde $p_{\oplus}$ é a proporção de exemplos positivos ($Y = +1$) e $p_{\ominus}$ é a proporção de exemplos negativos ($Y = -1$) em S. Definimos $0 \log_2 0 = 0$.

Tendo $E(S)$ definido, buscamos maximizar o ganho de informação GI para um determinado atributo A como

$$GI(S, A) = E(S) - \sum_{v \in V(A)} \frac{|Sv|}{|S|} E(S_v), \tag{2.9}$$

onde $V(A)$ é o conjunto de todos os valores possíveis para o atributo A e S_v é o subconjunto de S para o qual A tem o valor v. GI mede a redução esperada na entropia para um determinado atributo A. Quanto maior o ganho de informação GI de um atributo, melhor este atributo separa os dados [35].

Seleção do limiar apropriado para atributos contínuos

Quando trabalhamos com um conjunto S cujos atributos podem ter valores contínuos, temos outro problema a tratar: selecionarmos um limiar que transforme cada variável contínua em uma variável discreta.

Para cada atributo, temos que selecionar o limiar c que produza o maior ganho de informação. Para isso, ordenamos nossos exemplos segundo o valor da variável contínua sendo analisada e geramos um conjunto de limiares candidatos selecionando exemplos adjacentes que tenham classificação diferente. Para cada limiar candidato, dividimos o conjunto de treinamento segundo este limiar e calculamos o ganho de informação associado. Selecionamos o limiar que produz o maior ganho de informação. Aplicamos o procedimento recursivamente até atingirmos o critério de parada.

2.7.7 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) constituem atualmente um tema de grande interesse na comunidade de aprendizado de máquina criando um entusiasmo similar ao que as redes neurais artificiais despertou anos atrás.

A teoria geral dos *Support Vector Machines* foi apresentada por Cortes & Vapnik [7] em 1995 para classificação binária. Nessa abordagem, basicamente estamos procurando por um hiperplano de separação ótima entre duas classes. Isto é feito através da maximização da **margem**, menor distância entre um ponto pertencente à classe a e um ponto pertencente à classe b (Figura 2.6). Os pontos sobre as fronteiras são chamados **vetores de suporte** (*support vectors*), e o meio da margem é o nosso hiperplano de separação ótima. Para os pontos que estão do lado "errado" da margem discriminante, nós recalculamos os coeficientes de forma a reduzir sua influência (*soft margin method*). Quando não é possível achar um separador *linear*, os dados são projetados em um espaço de maior dimensão onde tornam-se linearmente separáveis. Essa projeção é feita através de técnicas

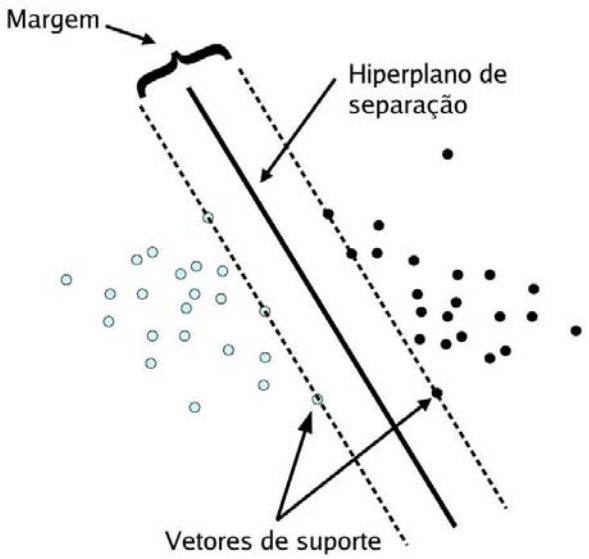


Figura 2.6: Classificação de um caso linearmente separável.

baseadas em *kernels* [46]. Um SVM pode ser dividido em três categorias: *linearmente separável*, *linearmente não-separável* e *não-linear*.

SVM linearmente separável

Suponha um problema de classificação de duas classes $Y(-1 = a \mid +1 = b)$. Denote a tupla (x_i, y_i), com $i = 1, \ldots, N$ sendo o nosso conjunto de treinamento. O vetor coluna x_i contém as características que nós achamos suficientes para distinguir os dois possíveis grupos e y_i é o resultado da classificação para x_i. Como exemplo, considere que o grupo com saída y_i = -1 representa uma imagem sem uma mensagem escondida e $y_i = +1$ representa uma imagem com uma mensagem escondida (*estego-imagem*). Com um SVM linearmente separável, queremos achar o hiperplano linear que separa as duas classes.

Vamos expressar a maximização da *margem* como uma função do vetor de pesos w e o viés (*bias*) b do hiperplano de separação. Assim, a distância entre um ponto x e um plano (w, b) é

$$\frac{w^T x + b}{||w||}, \tag{2.10}$$

onde $||.||$ é a norma Euclidiana[15] Como o hiperplano ótimo pode ter infinitas soluções devido às infinitas possibilidades de escalas de w e b, nós escolhemos a solução em que a

[15]A norma de um vetor X é dada por $||X|| = \sqrt{\langle X, X \rangle}$.

função discriminante se torna 1 para os exemplos de treinamento próximos da fronteira. Isto é conhecido como **hiperplano canônico**

$$|w^T x_i + b| = 1. \tag{2.11}$$

Desta forma, a distância dos exemplos mais próximos à fronteira é

$$\frac{|w^T x + b|}{||w||} = \frac{1}{||w||} \tag{2.12}$$

e a margem se torna

$$m = \frac{2}{||w||}. \tag{2.13}$$

O procedimento de um SVM linearmente separável é apresentado na Figura 2.7. O pro-

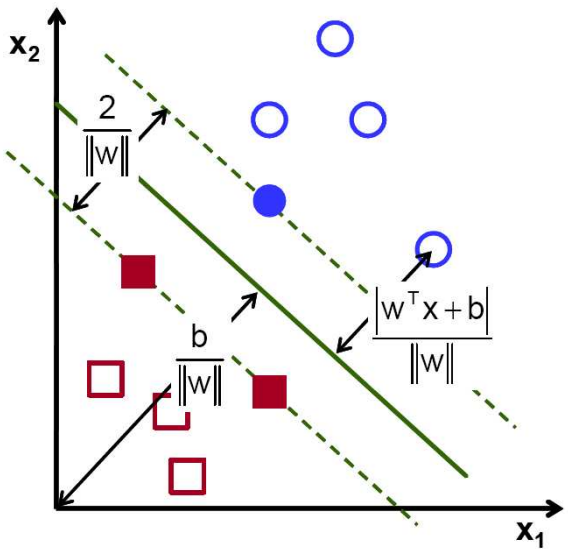


Figura 2.7: SVM linearmente separável

blema de maximizar a margem é equivalente a

$$\min J(w) = \frac{1}{2}||w||^2 \text{ sujeito a } y_i(w^T x_i + b) \geq 1 \ \forall i. \tag{2.14}$$

Por definição, $J(w)$ tem um mínimo global. Para resolver esta função, podemos empregar técnicas de otimização Lagrangiana. A introdução de multiplicadores de Lagrange para resolver este problema resulta em

$$L_P(w, b, \alpha_1, \ldots, \alpha_N) = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \alpha_i (w^t x_i + b) y_i + \sum_{i=1}^{N} \alpha_i, \tag{2.15}$$

onde α_i são os multiplicadores positivos de Lagrange. Esta função de erro precisa ser minimizada com relação a w e b. Como esse é um problema de otimização quadrática, uma solução para o problema dual L_D resulta na mesma solução para w, b e $\alpha_i, \ldots, \alpha_N$.

Definimos a **margem** para qualquer hiperplano dado como a soma das distâncias do hiperplano aos exemplares mais próximos das duas classes. Como dito antes, o hiperplano é escolhido de forma a maximizar a margem.

No problema dual L_D, a mesma função de erro L é maximizada em relação a α_i com a restrição de que as derivadas de L em relação a w e b sejam zero e que $\alpha_i \geq 0$. Diferenciando a Equação 2.15 com respeito a w e b e igualando a zero, temos as Equações 2.16 e 2.17

$$\frac{\partial L_P(w, b, \alpha)}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{N} \alpha_i x_i y_i, \tag{2.16}$$

$$\frac{\partial L_P(w, b, \alpha)}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{2.17}$$

Substituindo estas igualdades na Equação 2.15 sob a condição de otimalidade de Karush-Kuhn-Tucker [8, 19] $\partial J/\partial w = 0$, obtemos o problema dual

$$L_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j x_i^t x_j y_i y_j. \tag{2.18}$$

Os valores α_i da Equação 2.18 são usados para calcular a normal do hiperplano procurado.

Nós transformamos o problema de achar um ponto de inflexão (*saddle point*) para L_P no problema mais simples de maximização de L_D. Desta forma, a troca para a solução dual é recomendável. Outras razões para procedermos esta troca são: em L_P, problema primal, w tem um coeficiente para cada dimensão enquanto que em L_D, o problema dual, há uma solução para cada exemplo x_i apresentado (entrada x_i) ou seja L_D depende apenas dos multiplicadores Lagrangianos e não de (w, b).

A partir do hiperplano de separação (w, b), um novo exemplar z pode ser classificado simplesmente a partir do cálculo sobre qual lado do hiperplano ele está. Se o valor $w^t z + b \geq 0$ então o exemplar é classificado como $(\hat{y}_z) = +1$, caso contrário, ele é classificado como $(\hat{y}_z) = -1$.

SVM linearmente não-separável

Quando os dados não estão uniformemente dispostos em algum dos lados de um hiperplano, então um SVM linearmente separável não resultará em uma solução [19]. Podemos tratar tal situação relaxando as restrições iniciais através da introdução de **variáveis de folga** (*slack variables*) ξ_i que relaxam as restrições da equação do hiperplano canônico

$$y_i |w^T x_i + b| \geq 1 - \xi_i \; \forall i = 1 \ldots N. \tag{2.19}$$

O processo pode ser entendido como apresentado na Figura 2.8. Para $0 \leq \xi \leq 1$, os

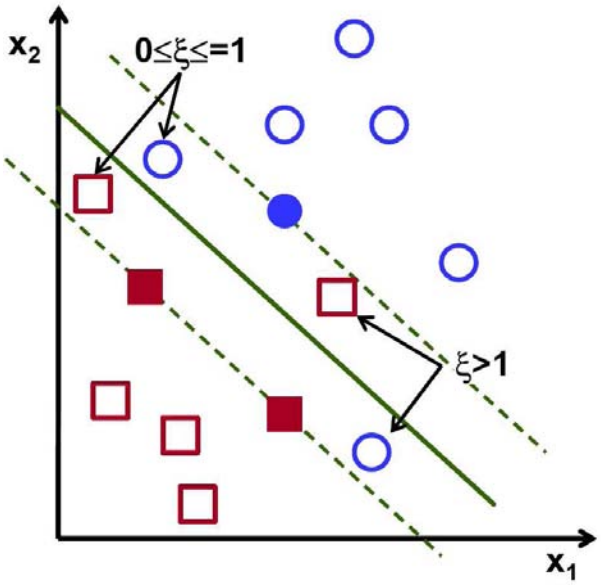


Figura 2.8: SVM não-linearmente separável

pontos (dados) estão do lado "correto" do hiperplano de separação mas dentro da região de máxima margem. Por outro lado, para $\xi > 1$ os pontos estão do lado "errado" do hiperplano de separação.

Com a introdução de variáveis de folga, o objetivo agora é achar o hiperplano de separação que produza a menor taxa de erros no treinamento, $\sum_i \xi_i$, enquanto maximiza a *margem*. A função de erro a ser minimizada é

$$\frac{||w||^2}{2} + C \sum_{i=1}^{N} \xi_i, \tag{2.20}$$

onde C é uma variável escalar escolhida pelo usuário. C é interpretado como um acordo (*tradeoff*) entre os erros de classificação e a capacidade do classificador. Valores altos para C implicam em poucos erros de classificação e maior complexidade no classificador. Por outro lado, valores baixos para C implicam preferência por soluções de baixa complexidade e, por conseqüência, com maior quantidade de erros na classificação. A minimização deste erro ainda é um problema de programação quadrática. Seguindo os passos da Seção 2.7.7, conseguimos expressar o problema dual (maximização) como

$$L_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \text{ sujeito a } \begin{cases} \sum_{i=1}^{N} \alpha_i y_i = 0, \\ 0 \leq \alpha_i \leq C \quad i = 1 \ldots N. \end{cases} \tag{2.21}$$

SVM não-linear

Uma característica dos SVMs vistos até o momento é que eles estão restritos a um *hiperplano linear*. Há casos em que um hiperplano linear não é capaz de garantir a classificação. Nestes casos, nós podemos empregar SVMs não lineares. Para atingir nosso objetivo, nós mapeamos os exemplos de treinamento num espaço Euclidiano de mais dimensões (possivelmente milhões). Podemos denotar este mapeamento por

$$\phi : \mathcal{L} \rightarrow \mathcal{H}, \tag{2.22}$$

que mapeia nosso conjunto de treinamento do espaço dimensional Euclidiano $\mathcal{L}$ para o espaço de maior dimensão $\mathcal{H}$. Substituindo x_i por $\phi(x_i)$ em todo o conjunto de treinamento, passamos a ter um SVM no espaço dimensional $\mathcal{H}$. Infelizmente, pode ser inconveniente trabalhar em um espaço tão grande quanto $\mathcal{H}$ possa ser. No entanto, o erro a ser maximizado (Equação 2.21) depende apenas de produtos internos sobre exemplares do conjunto de treinamento, $x_i^t x_j$. Assim, podemos definir uma função *kernel* K capaz de fazer este mapeamento sob demanda (*on-the-fly*)

$$K(x_i, x_j) = \phi(x_i)^t \phi(x_j). \tag{2.23}$$

Há várias escolhas possíveis para K, dentre elas funções radiais de base (RBF - *radial basis functions*) ou mesmo polinomiais (PLN) [19]. Substituindo $\phi(x_i)^t\phi(x_j)$ com a função *kernel* $K(x_i, x_j)$, temos um SVM no espaço dimensional $\mathcal{H}$ com um mínimo impacto computacional quando comparado com o espaço dimensional $\mathcal{L}$.

Um novo exemplar z, é classificado pela determinação sobre qual lado do hiperplano de separação (w, b) ele está. Especificamente, se o valor $w^t\phi(z) + b \geq 0$ ele é classificado como da classe $\hat{y}_z = +1$, caso contrário como da classe $\hat{y}_z$ = -1. No entanto, este teste é ainda não prático e precisamos também testar z no espaço $\mathcal{H}$ usando a mesma função K

$$w^t\phi(z) + b = \sum_{i=1}^{N} \alpha_i K(x_i, z) y_i + b. \tag{2.24}$$

2.7.8 Coletâneas

Nesta abordagem, também conhecida como *Bootstrap aggregation* ou (*Bagging*) nós avaliamos as predições sobre uma coleção de amostras (*bootstrap samples*)[16] para reduzir sua variância e conseqüentemente o erro na predição [19].

Nós criamos uma coletânea (*ensemble*) a partir do treinamento individual de classificadores nas amostras feitas do conjunto de dados. Seja X, o conjunto de dados de entrada

[16] *Bootstrap samples* são amostras com reposição e mesmo tamanho feitas a partir de um conjunto de dados.

e Z^i, $i = 1, 2, \ldots, B$, uma amostra de X. Para procedermos a classificação em cada Z^i, nós selecionamos um classificador (LDA, CTREE, SVM, entre outros). Normalmente, aplicamos o mesmo classificador em todas as amostras Z^i.

Em cada passo, nós construímos um conjunto de treinamento fazendo amostras com reposição a partir do conjunto de dados original. Como resultado desta operação, cada classificador é treinado em média sobre 63.2% dos exemplos do conjunto de treinamento [19].

Para cada elemento x, armazenamos a predição $\vec{f^i}(x)$. Definimos a estimativa da coletânea para x, como

$$\vec{f}_{bag}(x) = \frac{1}{B}\sum_{i=1}^{B}\vec{f^i}(x). \tag{2.25}$$

A Equação 2.25 funciona como uma votação dentro da coletânea. Ao avaliarmos todas as classificações feitas para x, nós o classificamos como pertencente à classe para a qual ele foi classificado mais vezes. A Figura 2.9 apresenta um exemplo de como fazemos o treinamento e classificação utilizando a abordagem *Bagging*.

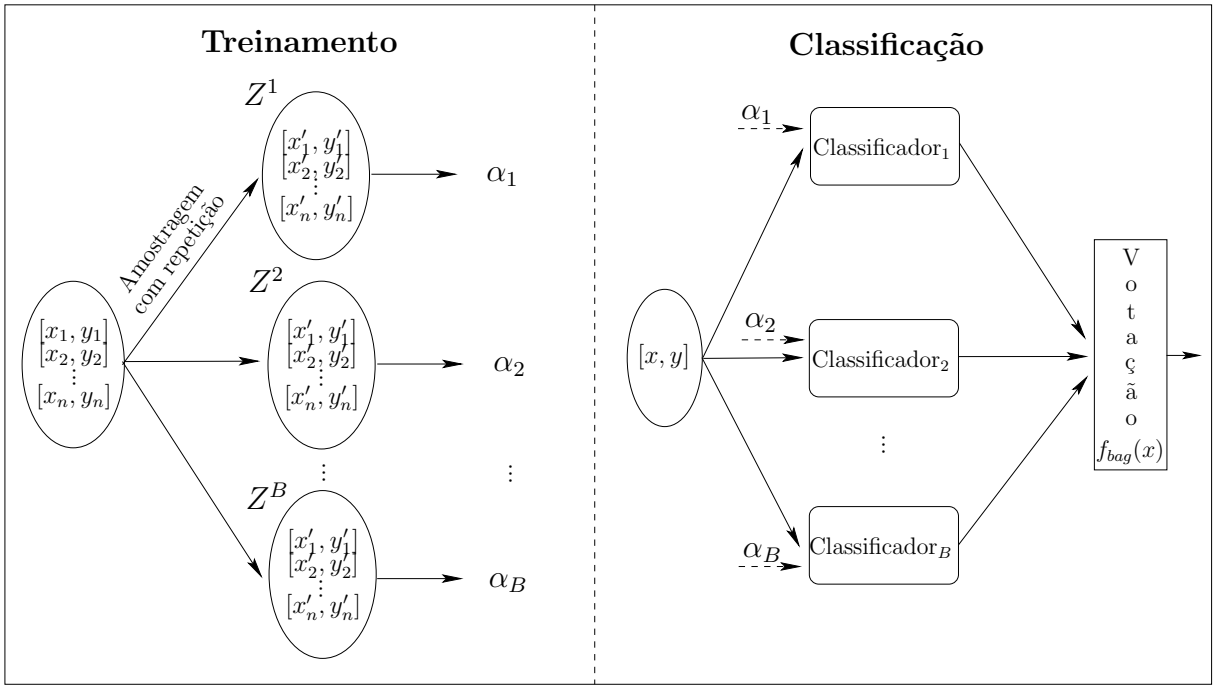


Figura 2.9: Treinamento e classificação utilizando *Bagging*.

Armazenamos os coeficientes relativos a cada classificador aplicado (α) de modo que ao recebermos um exemplar a ser testado, submetemos este exemplar aos B classificadores e procedemos a votação.

2.8 Considerações finais

Neste capítulo descrevemos o estado da arte da esteganografia e da esteganálise. A seguir, no Capítulo 3, apresentamos nossa metodologia de detecção de mensagens escondidas em imagens digitais. Descrevemos a seleção de regiões características, análise dos descritores estatísticos χ^2 e U_T e o estágio da randomização progressiva. Apresentamos também como reunir estas características e utilizar o aprendizado de máquinas para classificar se uma dada imagem possui ou não uma mensagem escondida em seus *bits* menos significativos.

Capítulo 3

Randomização Progressiva para esteganálise

Decifra-me (detecta-me) *ou te devoro!*
(O Enigma da Esfinge)

Neste capítulo, nós apresentamos nossa metodologia de detecção de mensagens escondidas em imagens digitais. Esta metodologia consiste em três estágios: seleção de regiões características, análise dos descritores estatísticos χ^2 e U_T e o estágio da randomização progressiva.

3.1 Descritores estatísticos

Qualquer procedimento de mascaramento nos canais LSB altera o conteúdo de um conjunto de *pixels* selecionados. Isto implica em uma mudança nas estatísticas dos valores dos *pixels* em uma vizinhança local.

Em um canal de cores com L *bits* de profundidade, nós podemos representar 2^L valores possíveis. Se nós dividirmos estes valores em 2^{L-1} pares que apenas se diferenciam no canal LSB, nós consideramos todos os padrões de vizinhança possíveis para os LSBs. Denominamos cada um destes pares como *pares de valor* (*pair of value*) ou PoV na seqüência [54].

Quando nós utilizamos todos os LSBs disponíveis para esconder uma mensagem em uma imagem, a distribuição dos valores par e ímpar de um PoV tem a mesma freqüência 0/1 da distribuição dos *bits* da mensagem. O objetivo da análise estatística é comparar a distribuição de freqüência esperada e a distribuição de freqüência observada nos PoVs [54]. Entretanto, nós não temos a imagem original e, desta forma, a freqüência esperada.

A função de mascaramento afeta apenas os LSBs deixando a distribuição dos PoVs inalterada após o mascaramento. Assim, nós supomos que a imagem recebida tem uma mensagem pseudo-aleatória escondida que ocupe todo o canal LSB fazendo com que a proporção de 0s e 1s no canal LSB seja próximo de 1 para 1. A partir desta suposição, temos que a freqüência esperada é a média aritmética das duas freqüências em cada PoV.

Quando nós aplicamos um descritor estatístico que mede a diferença entre as freqüências esperada e observada, temos duas possibilidades: (1) a imagem recebida não é uma estego-imagem resultando em um valor alto no descritor, dado que a suposição era de que esta era uma estego-imagem ou (2) a imagem é uma estego-imagem e a suposição foi correta resultando em valores baixos do descritor.

Como exemplo, considere um PoV igual ao par ($p_1 = 100$, $p_2 = 101$) na base 2. Considere a freqüência observada igual a $p_1 = \frac{20}{60}$ *pixels* e $p_2 = \frac{40}{60}$ *pixels*. Isto resulta 60 *pixels* para um mascaramento completo neste PoV. Considerando um mascaramento fictício de uma mensagem de 60 *bits* (um *bit* por *pixel* selecionado), sendo metade de *bits* 0 e metade de *bits* 1, temos um PoV resultante com a distribuição $p_1 = \frac{30}{60}$ *pixels* e $p_2 = \frac{30}{60}$ *pixels* (média aritmética entre p_1 e p_2). Dado que as operações de mascaramento afetam apenas os LSBs ocasionando uma redistribuição dos valores dentro do mesmo PoV, consideramos esta segunda freqüência como nossa freqüência esperada.

Como apresentado em [54, 42], nós podemos aplicar o teste do χ^2 e U_T (Teste universal de Ueli Maurer) sobre estes PoVs para detectar mensagens escondidas em imagens digitais. A partir das freqüências encontradas, nós calculamos os descritores como apresentado na Seção 2.5. A probabilidade pr de mascaramento é dada pelo complemento da função de distribuição acumulativa

$$pr = 1 - \int_0^{\chi^2} \frac{t^{(\nu-2)/2} e^{-t/2}}{2^{\nu/2}\Gamma(\nu/2)} dt, \tag{3.1}$$

onde Γ é a função Euler-Gama. Podemos calcular esta probabilidade em diferentes partes da imagem e com diferentes janelas (regiões). Em nossa abordagem, nós analisamos os relacionamentos dos descritores. Não analisamos pr.

De forma geral, para uma imagem que não contenha qualquer informação escondida, é esperada que a probabilidade de mascaramento seja próximo de zero em qualquer amostragem [42]. Em termos de χ^2, é esperado um valor alto. Quanto maior a mensagem escondida na imagem, menor o valor χ^2, ou seja, maior a condição de randomicidade do canal LSB da imagem analisada.

Em nossa metodologia de detecção, nós estendemos estes dois descritores. Abordagens anteriores [54, 42] que utilizam estes descritores apenas detectam mensagens seqüencialmente escondidas a partir dos primeiros LSBs disponíveis. Estas abordagens consideram apenas os valores diretos dos descritores e não levam em consideração que o valor mínimo para detecção de uma mensagem pode ser muito distinto e dependente do contexto da

imagem sendo analisada. A Figura 3.1 apresenta, em quatro imagens, a fragilidade da interpretação direta do descritor χ^2 e U_T[1].

O valor mínimo para detecção da mensagem é dependente do contexto da imagem sendo analisada. Não podemos dizer, por exemplo, que imagens com o descritor $\chi^2 \leq 160.0$ ou $U_T \geq 6.49$ possuem mensagens escondidas.

A medição direta dos descritores constitui uma medida de estatística de baixa ordem. Esta abordagem pode ser superada por técnicas que mantém perfis estatísticos básicos (média e desvio-padrão)[2] no processo de mascaramento. Nós tratamos este problema avaliando o comportamento de nossos descritores ao longo de regiões características selecionadas.

3.2 Seleção de regiões características

A partir de uma imagem I, nós queremos r regiões com tamanho $l \times l$ *pixels* que contenham informações suficientes para produzir bons descritores já que existem algumas técnicas de mascamento de mensagens que buscam regiões na imagem com maior riqueza de detalhes de modo a reduzir os artefatos inseridos no canal LSB durante o mascaramento [23, 53].

Dado que afirmamos que nossa metodologia de detecção de mensagens é independente da técnica de esteganografia utilizada, detecção cega (*blind detection*), nosso arcabouço de detecção precisa ser capaz de detectar mensagens aleatoriamente ou seqüencialmente escondidas na imagem, bem como mensagens escondidas em regiões específicas que exploram alguma característica de vizinhança da imagem de cobertura, como por exemplo a riqueza de detalhes em certas regiões.

A seleção das regiões de nosso interesse consiste em duas etapas (1) nós selecionamos quatro regiões que cubram a imagem inteira sem sobreposição, Q_{rs} ou *regiões de Quads*, (2) nós identificamos quatro regiões na imagem que são ricas em detalhes, H_{rs} ou *regiões de Harris*. Todas estas regiões são mostradas para um exemplo real na Figura 3.2.

Para acharmos as regiões de Harris, nós utilizamos um filtro como definido por Harris e Stephens [21],

$$H_{rs} = \det(G) - \alpha \ \mathrm{tr}(G)^2, \tag{3.2}$$

onde det(.) é o determinante, tr(.) é o traço, α é um fator de escala e G é uma matriz

[1] 3.1(a) Afresco *The Burning of the Borgo* de Raphael, Stanza dell'Incendio – Palácio do Vaticano, Roma. 3.1. (b) Tela *Portrait of Nicolaes Ruts* de Rembrandt van Rijn, The Frick Collection, New York.

[2] A manutenção do perfil estatístico da imagem original consiste em alterar LSBs para cada *bit* escondido de modo a manter média, desvio-padrão, energia, contraste entre outras medidas, do canal LSB o mais próximas possíveis da imagem de cobertura.

(a) $\chi^2 = 187.72$ $U_T = 6.48$

(b) $\chi^2 = 752.44$ $U_T = 6.13$

(c) $\chi^2 = 151.16$ $U_T = 6.49$

(d) $\chi^2 = 403.58$ $U_T = 6.48$

Figura 3.1: Fragilidade da interpretação direta dos descritores χ^2 e U_T. (a) e (b) Imagens sem conteúdo escondido. (c) e (d) imagens com mensagens escondidas de tamanho equivalente a 25% dos LSBs disponíveis.

simétrica 2×2

$$G = \begin{bmatrix} \sum \nabla_x^2 & \sum \nabla_x \nabla_y \\ \sum \nabla_x \nabla_y & \sum \nabla_y^2 \end{bmatrix}. \tag{3.3}$$

A intuição do filtro de Harris é que ele retorna respostas altas em regiões ricas em texturas. Aplicamos esse filtro em regiões (janelas w) igualmente espaçadas na imagem. Um tamanho típico é 7×7 *pixels*. Para cada janela w, nós calculamos o gradiente no eixo x e no eixo y e calculamos G e H_{rs}. Criamos uma matriz de pontos H_{rs} onde cada ponto corresponde à posição central de w. Ao final de todo o processo, temos um valor de H_{rs} para cada ponto da imagem exceto para os pontos de borda. Selecionamos os quatro pontos da maior magnitude e traçamos uma região em torno de cada um destes pontos tendo os mesmos como pontos centrais da região.

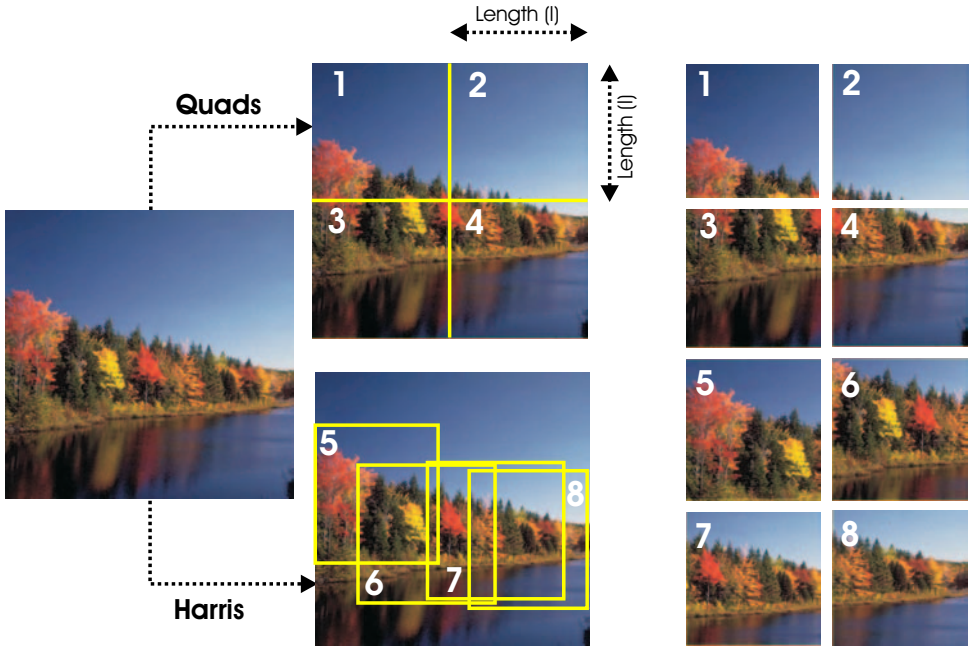


Figura 3.2: Extração das regiões Q_{rs} e H_{rs}.

3.3 Randomização progressiva

Neste trabalho, nós introduzimos uma nova metodologia para extrair descritores robustos à técnicas de mascaramento de mensagens nos canais LSB de imagens digitais.

A metodologia de **Randomização Progressiva** (PR), consiste na progressiva aplicação de transformações de mascaramento nos LSBs de uma imagem. Ao receber uma imagem I como entrada, o método cria n imagens, que apenas se diferenciam da imagem original no canal LSB. A saída (O_i) é uma transformação direta da entrada $O_i = T_i(I)$.

As T_i transformações representam possíveis processos de mascaramento com mensagens de tamanhos diferentes. Em nossos experimentos (Cap. 4), nós utilizamos $n = 6$ com tamanhos de mensagem[3] 1%, 5%, 10%, 25%, 50% e 75%. Dado que estamos detectando mensagens aleatoriamente escondidas na imagem de cobertura, quanto maior o tamanho da mensagem escondida, maior a entropia no canal LSB.

Para a imagem original e para cada imagem gerada, nós calculamos os valores dos descritores estatísticos escolhidos nas regiões características selecionadas. Em nossos experimentos (Cap. 4), nós selecionamos oito regiões características e dois descritores estatísticos (χ^2 e U_T).

[3]Uma mensagem com tamanho m% é um bloco de informação que utiliza m por cento dos LSBs disponíveis na imagem de cobertura.

Em nossa metodologia, nós precisamos considerar variações no contexto de diferentes imagens. Nossos descritores precisam ser robustos a essas variações. Nós estamos interessados nas taxas de variação de nossos descritores ao invés de seus valores em si. Normalizando todas as medidas em relação à imagem original de entrada I, nós tratamos este problema. Desta forma, nós tomamos a razão relativa de cada passo da randomização progressiva e a imagem original,

$$Norm(O_i) = d_j(O_i)/d_j(I), \tag{3.4}$$

onde d denota um descritor de uma imagem tomado em uma região $1 \leq j \leq k$. Em nossa abordagem, o descritor d pode ser χ^2 ou U_T. A Figura 3.3 apresenta o comportamento de nossos descritores para uma imagem ao longo da randomização progressiva. Quanto maior a mensagem escondida (eixo x), maior é o valor normalizado do descritor (eixo y).

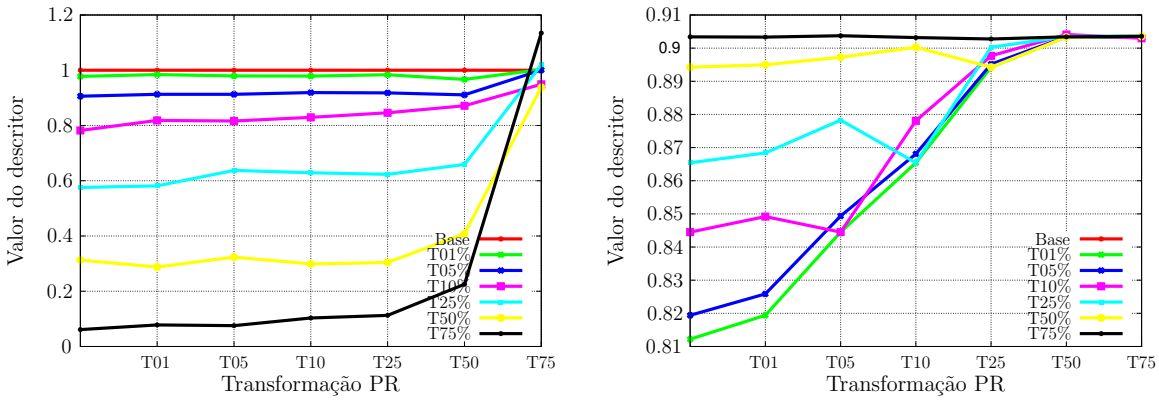


Figura 3.3: Comportamento dos descritores normalizados sobre a região característica Q_1 da Figura 3.2 ao longo da Randomização Progressiva. (a) χ^2. (b) U_T.

Após a normalização, o comportamento dos descritores torna-se independente do contexto da imagem analisada. As Figuras 3.4(a-f) apresentam as diferenças entre os descritores de uma imagem sem mensagem escondida e uma imagem com uma mensagem escondida de tamanho $|M| = 25\%$ dos LSBs disponíveis.

Após o mascaramento de uma mensagem de tamanho $|M| = 25\%$ dos LSBs disponíveis, o valor relativo entre cada etapa da randomização progressiva e a imagem sendo analisada torna-se menor. Ao analisarmos este comportamento para um conjunto de imagens e utilizarmos aprendizado supervisionado, somos capazes de diferenciar entre uma imagem que nunca recebeu uma mensagem escondida e uma imagem que possui uma mensagem escondida dado que a randomização progressiva tem um comportamento distinto em cada uma das situações.

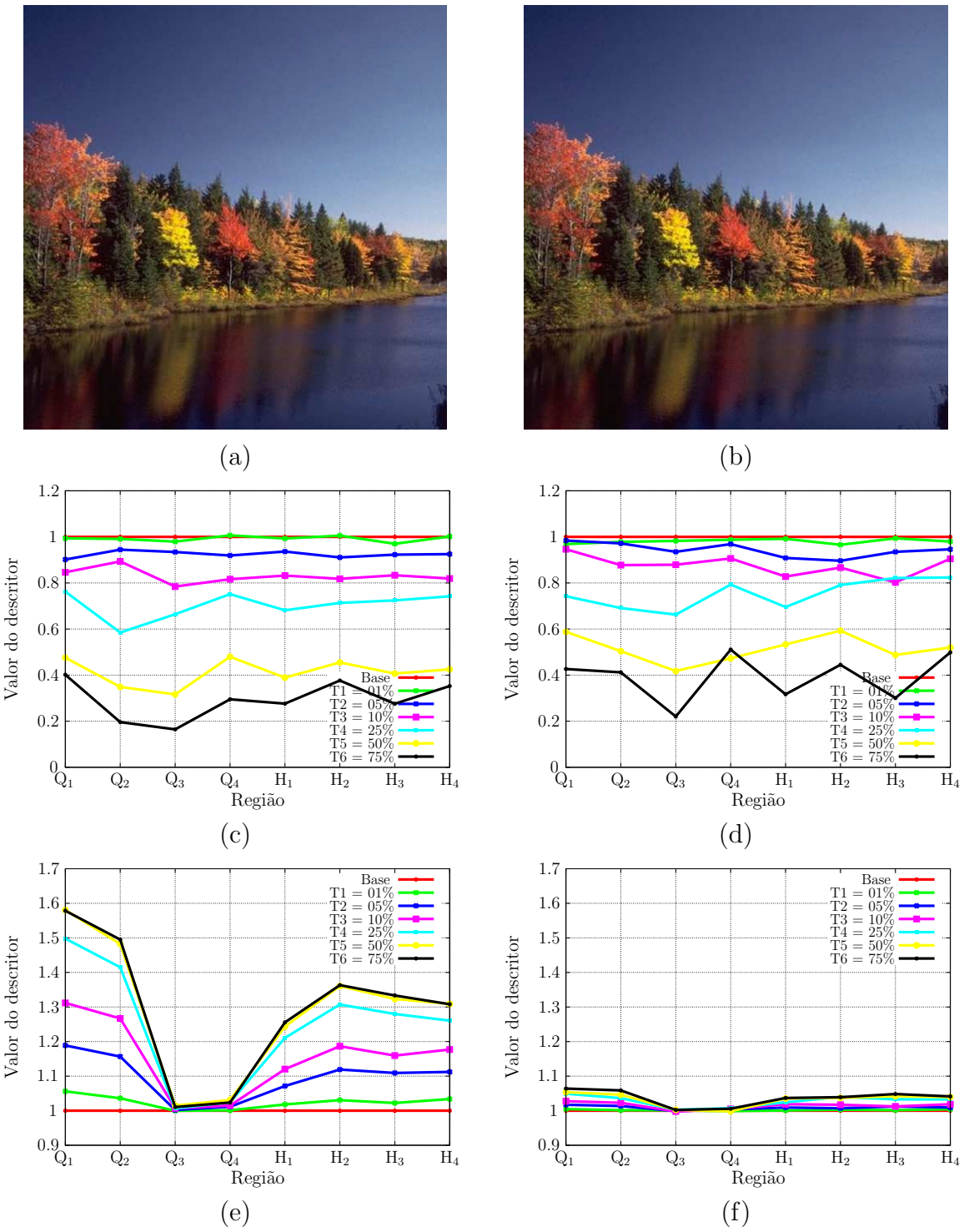


Figura 3.4: Comportamento dos descritores normalizados sobre uma imagem ao longo da randomização progressiva. (a) Imagem antes do mascaramento. (b) Imagem após o mascaramento. (c) χ^2 antes do mascaramento. (d) χ^2 após um mascaramento de $|M| = 25\%$ dos LSBs disponíveis. (e) U_T antes do mascaramento. (f) U_T após um mascaramento $|M| = 25\%$ dos LSBs disponíveis.

Resumimos o procedimento de construção do vetor característico a partir de uma imagem de entrada I como:

Construção do vetor característico

1. **Aplique** a randomização progressiva sobre I. Isto resulta em n imagens transformadas mais a imagem original;
2. **Calcule** as m regiões características (e.g. $m = 8$, 4 Q_{rs} e 4 H_{rs}) para cada imagem;
3. **Calcule** os k descritores (e.g. $k = 2$, χ^2 e U_T) para cada região característica totalizando $k \times m \times (n + 1)$ valores de descritor (e.g. $2 \times 8 \times 7 = 112$);
4. **Normalize** o conjunto de descritores relativo a uma etapa da randomização progressiva em relação à imagem I. Isto é, normalize cada valor de descritor d_i medido em uma determinada região de uma etapa da randomização progressiva em relação à mesma região medida sobre a imagem I. A normalização resulta em $k \times m \times n$ valores de descritor (e.g. $2 \times 8 \times 6 = 96$).

Resumimos o procedimento de classificação como:

Treinamento e classificação

1. **Selecione** um conjunto de imagens para treinamento;
   - **Construa** o vetor característico $R^{k \times m \times n}$ para cada imagem (e.g. em nossa abordagem temos $R^{2 \times 8 \times 6} = R^{96}$ dimensional);
   - **Selecione** um classificador;
   - **Proceda** o treinamento do classificador utilizando o conjunto de vetores característicos;
   - **Armazene** os coeficientes de treinamento;
2. Ao receber uma imagem de teste I', **construa** o vetor característico de I';
3. **Forneça** o vetor característico de I' ao classificador para conseguir uma inferência a respeito de I'. Utilize para isso, os coeficientes de treinamento previamente calculados.

Capítulo 4

Experimentos e validação

Reason, observation, and experience; the holy trinity of science.[1]
(Robert Green Ingersoll)

Neste capítulo, nós descrevemos o treinamento e validação de nosso arcabouço de detecção de mensagens. Apresentamos a exatidão de nosso sistema com respeito aos classificadores selecionados e comparamos nossos resultados com trabalhos anteriores na literatura [13, 31, 42, 54].

Nós validamos nossa metodologia em um banco de dados com 20.000 imagens reais. Todas as imagens têm uma resolução de 512×512 *pixels* e são armazenadas no formato PNG [52]. Estas imagens são livres de *copyright* e vieram de bancos de imagens individuais e da *internet*.

4.1 Treinamento e teste

Nós assumimos que todas as nossas 20.000 imagens são imagens sem conteúdo escondido (não-estego). Para o procedimento de classificação, nós precisamos de exemplares que representem a classe de estego-imagens e de exemplares que representem a classe não-estego. Uma estego-imagem pode conter uma mensagem escondida de tamanho variável. Nós selecionamos $n = 6$ possíveis tamanhos de mensagens 1%, 5%, 10%, 25%, 50% e 75% dos LSBs disponíveis para simular estego-imagens.

Nós criamos uma versão do banco de imagens contendo estego-imagens para cada um dos seis tamanhos relativos de mensagens escolhidos, aplicamos a Randomização Progressiva em cada imagem e treinamos um classificador de duas classes (não-estego/estego) para cada grupo de estego-imagens.

[1] Razão, observação e experiência; a trindade sagrada da ciência.

4.2 Validação

Nós selecionamos oito regiões (Seção 3.2), quatro regiões (Quads) espacialmente constantes e quatro regiões (Harris) dependentes do contexto da imagem. Para cada região, nós calculamos dois descritores estatísticos (χ^2 e U_T). Isto nos resulta $2 \times 8 = 16$ valores de descritor para cada etapa da randomização progressiva. Utilizando $n = 6$ transformações possíveis, nós temos seis imagens resultantes da randomização progressiva mais a imagem original de entrada o que resulta sete imagens. Assim, temos sete imagens, oito regiões características por imagem e dois descritores estatísticos por região, o que resulta $7 \times 8 \times 2 = 112$ valores de descritor. Após a normalização (Seção 3.3), nossos procedimentos de classificação operam em um espaço 96-dimensional, dado que a normalização dos valores de descritor da imagem original I com ela mesma resulta em coeficientes que não são úteis aos nossos procedimentos de classificação.

Nossa implementação em C++, executando em um AMD 64 bits 3200+ com 2 GB de memória RAM, gera o vetor 96-dimensional de uma imagem com resolução de 512×512 *pixels* em 30 segundos.

Após treinarmos os classificadores selecionados (Seção 2.7), um para cada tamanho relativo de mensagem, nós testamos nossa metodologia como descrito na Seção 4.1. Em nossos experimentos, nós utilizamos o pacote de *software* R [51] para treinarmos e avaliarmos os diferentes classificadores selecionados.

Em todas as nossas análises, nós aplicamos validação cruzada (Seção 2.7.3) com 10 partições de mesmo tamanho a partir do conjunto de análise. Em todas as tabelas subseqüentes, nós reportamos os resultados utilizando as mesmas partições.

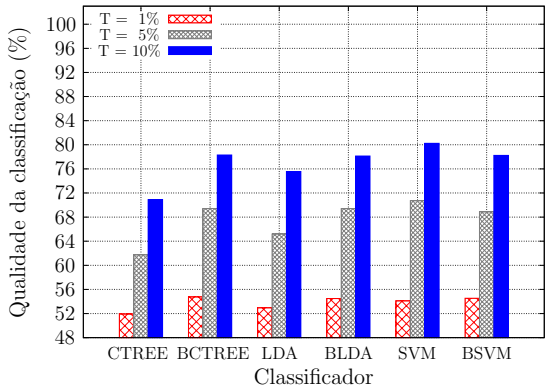
4.3 Randomização progressiva

A Tabela 4.1 e as Figuras 4.1(a-b) apresentam os resultados da Randomização Progressiva para os classificadores selecionados. Na Tabela 4.1 apresentamos os classificadores individuais em fundo branco e, em fundo cinza, as suas respectivas versões utilizando a coletânea *Bagging*. Escolhemos 37 iterações para o *Bagging* como explicado na Seção 4.3.2. Os valores percentuais indicam a qualidade da classificação. Uma qualidade de 100% indica que a classificação foi corretamente efetuada em 100% dos exemplos fornecidos para teste.

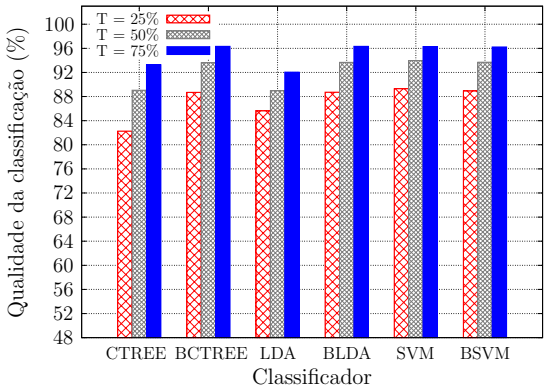
O classificador SVM-RBF individual produz o melhor resultado, independentemente do tamanho do conteúdo escondido. Contudo, SVM-RBF é um classificador computacionalmente caro e de difícil implementação. Neste caso, podemos utilizar a abordagem *Bagging* com um classificador mais fraco como o LDA ou CTREE.

	CTREES		**LDA**		**SVM-RBF**		**Tipo**
	μ	σ	μ	σ	μ	σ	
01%	51,9%	1,4%	52,9%	1,6%	54,1%	0,9%	Individual
	54,8%	1,4%	54,5%	1,0%	54,5%	1,0%	Bagging
05%	61,7%	0.8%	65,2%	1,0%	70,7%	0,9%	Individual
	69,4%	0,9%	69,4%	0,8%	69,0%	0,8%	Bagging
10%	70,9%	0,8%	75,5%	0,7%	80,2%	0,5%	Individual
	78,3%	0,4%	78,1%	0,8%	78,2%	0,7%	Bagging
25%	82,3%	0,6%	85,6%	0,8%	89,3%	0,6%	Individual
	88,7%	0,4%	88,7%	0,4%	88,9%	0,6%	Bagging
50%	89,5%	0,7%	89,0%	0,6%	94,0%	0,5%	Individual
	93,6%	0,5%	93,7%	0,5%	93,7%	0,5%	Bagging
75%	93,3%	0,4%	92,0%	0,6%	96,3%	0,3%	Individual
	96,3%	0,3%	96,3%	0,4%	96,2%	0,4%	Bagging

Tabela 4.1: Classificação utilizando quatro Q_{rs} e quatro H_{rs}. μ and σ são referentes à validação cruzada.



(a) $|M| \in \{1\%,\ 5\%,\ 10\%\}$ dos LSBs.



(b) $|M| \in \{25\%,\ 50\%,\ 75\%\}$ dos LSBs.

Figura 4.1: Classificação utilizando quatro Q_{rs} e quatro H_{rs}.

Utilizando Bagging e LDA, conseguimos a abordagem de classificação com melhor custo/benefício, produzindo bons resultados com uma baixa complexidade computacional. Com *Bagging* e LDA, nós conseguimos diferenciar uma estego-imagem de uma imagem sem conteúdo escondido com uma precisão de $\mu = 88,7\%$ e $\sigma = 0,4\%$ para mensagens de tamanho relativo $|M| = 25\%$ dos LSBs disponíveis. Isto é estatisticamente próximo da abordagem computacionalmente mais cara SVM-RBF (e.g. $\mu = 89,3\%$ e $\sigma = 0,6\%$). Nossos resultados indicam claramente que o classificador SVM-RBF não é beneficiado pela abordagem *Bagging* já que utiliza no cálculo no procedimento de classificação apenas os elementos próximos à fronteira de decisão (margem).

4.3.1 Influência das regiões de Harris

Nós utilizamos as regiões de Harris em uma tentativa de achar mascaramentos localizados que um indivíduo pode fazer em áreas de grandes detalhes em uma imagem[2].

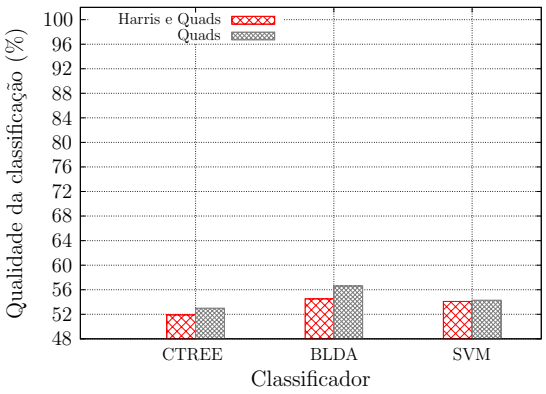
É importante entendermos os impactos positivos e negativos ao utilizarmos as regiões de Harris quando a mensagem sendo escondida é igualmente distribuída por toda a imagem, como a que utilizamos para classificar e validar nossos experimentos em nosso trabalho.

Utilizando quatro regiões de Harris e quatro regiões de Quads, temos uma qualidade de classificação inferior do que quando utilizamos oito regiões constantes igualmente distribuídas pela imagem. Entretanto, quanto maior a mensagem escondida, menor é a diferença nas qualidades de classificação de ambos os métodos. A Tabela 4.2 e as Figuras 4.2(a-f) mostram uma comparação entre estas duas possibilidades. Na Tabela 4.2, mostramos em fundo cinza a abordagem de detecção utilizando oito regiões igualmente distribuídas pela imagem e, em fundo branco, mostramos a abordagem utilizando quatro regiões de Harris e quatro regiões de Quads.

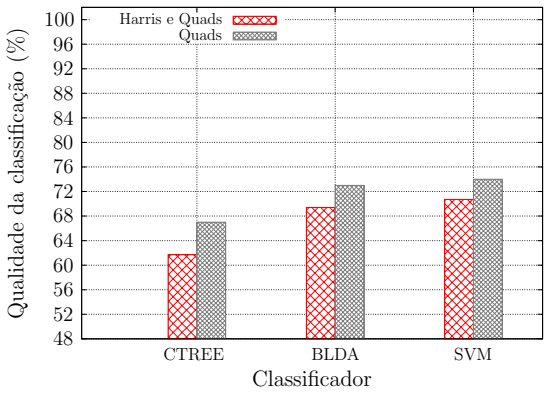
[2]Em tais áreas, os artefatos inseridos são menos perceptíveis [23, 53].

	CTREES		**BLDA**		**SVM-RBF**		**Tipo**
	μ	σ	μ	σ	μ	σ	
01%	51,9%	1,4%	54,5%	1,0%	54,1%	0,9%	Harris-Quads
	53,02%	0,8%	56,6%	0,9%	54,3%	1,0%	Quads
05%	61,7%	0.8%	69,4%	0,8%	70,7%	0,9%	Harris-Quads
	67,0%	0,6%	73,2%	0,5%	74,0%	0,4%	Quads
10%	70,9%	0,8%	78,1%	0,8%	80,2%	0,5%	Harris-Quads
	74,5%	0,5%	82,2%	0,3%	83,3%	0,7%	Quads
25%	82,3%	0,6%	88,7%	0,4%	89,3%	0,6%	Harris-Quads
	85,6%	0,6%	91,6%	0,5%	90,9%	0,5%	Quads
50%	89,5%	0,7%	93,7%	0,5%	94,0%	0,5%	Harris-Quads
	92,3%	0,6%	95,9%	0,4%	95,0%	0,4%	Quads
75%	93,3%	0,4%	96,3%	0,4%	96,3%	0,3%	Harris-Quads
	95,4%	0,5%	97,3%	0,5%	96,4%	0,3%	Quads

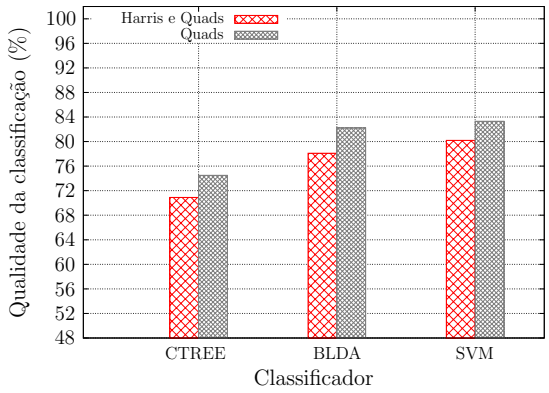
Tabela 4.2: Classificação utilizando quatro Q_{rs} e quatro H_{rs} *vs.* oito Q_{rs}. μ and σ são referentes à validação cruzada.



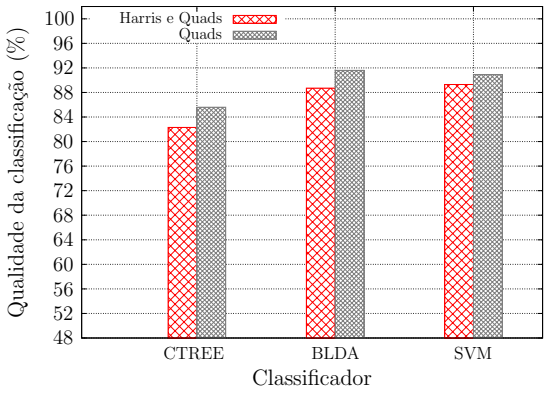
(a) $|M| = 1\%$ dos LSBs.



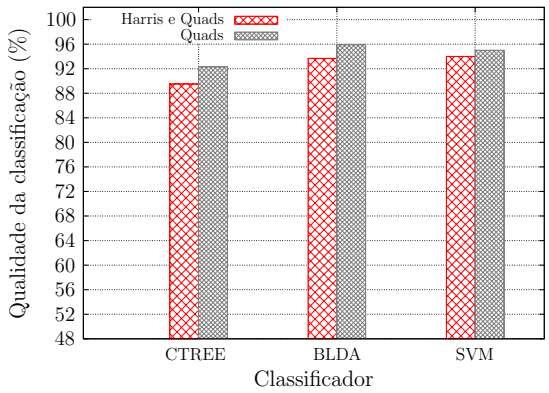
(b) $|M| = 5\%$ dos LSBs.



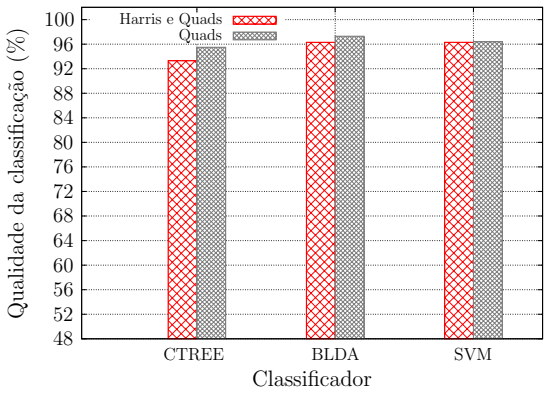
(c) $|M| = 10\%$ dos LSBs.



(d) $|M| = 25\%$ dos LSBs.



(e) $|M| = 50\%$ dos LSBs.

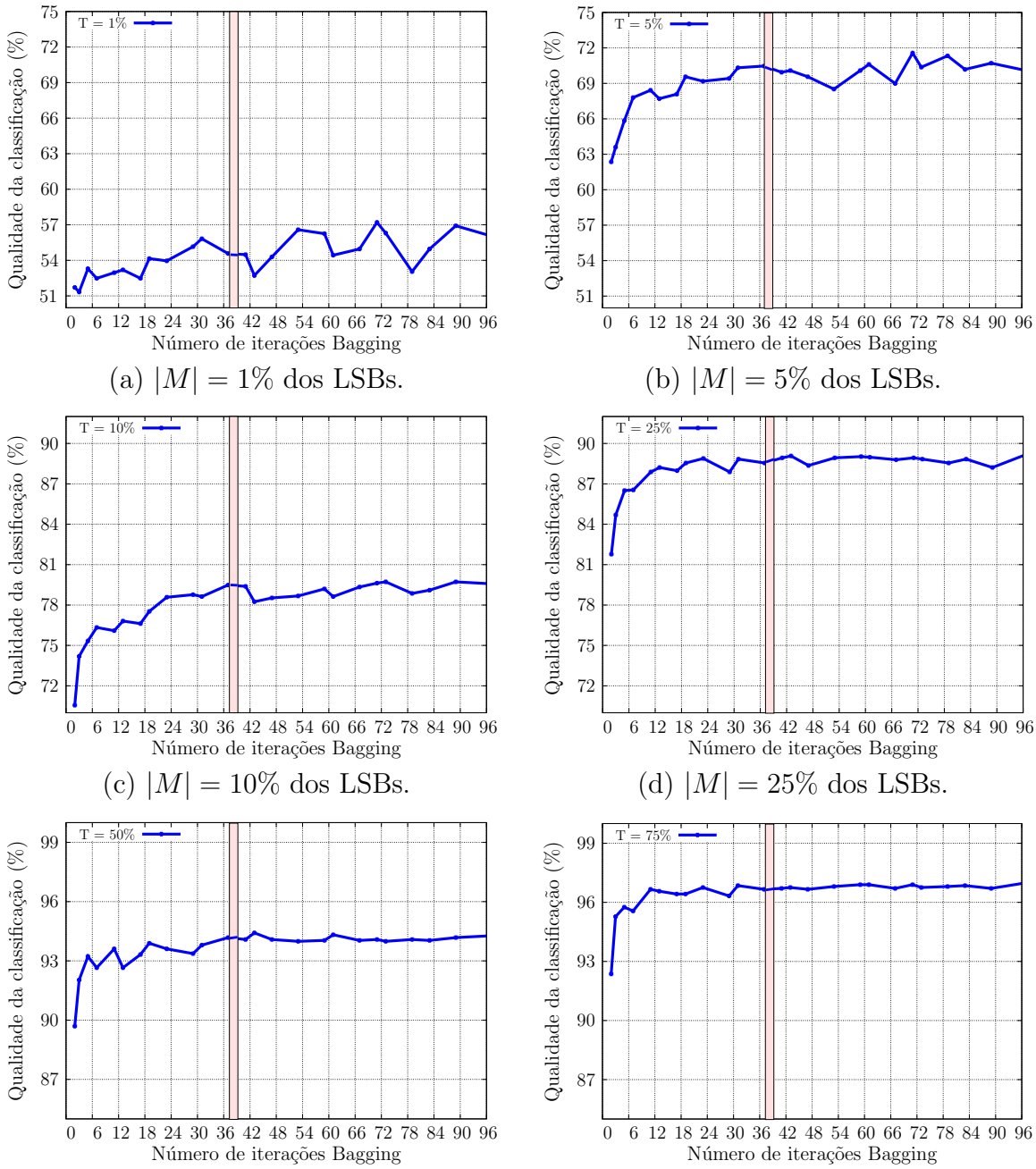


(f) $|M| = 75\%$ dos LSBs.

Figura 4.2: Classificação utilizando quatro Q_{rs} e quatro H_{rs} *vs.* oito Q_{rs}.

4.3.2 Número ideal de iterações utilizando Bagging

Para procedermos a escolha do melhor número de iterações para o classificador *Bagging*, nós consideramos a relação da qualidade da classificação obtida e o tempo necessário para procedermos a classificação. Embora o tempo de classificação seja linear em relação ao número de iterações utilizadas, não podemos simplesmente aumentar indefinidamente o número de iterações do classificador *Bagging* e esperar uma classificação quase perfeita. As Figuras 4.3(a-f) mostram o comportamento da qualidade da classificação do classificador LDA utilizando *Bagging*. Escolhemos 37 iterações como o melhor custo/benefício entre tempo e qualidade de classificação. Procuramos escolher um valor comum aos seis tamanhos relativos de mensagens utilizados em nossos testes.



(a) $|M| = 1\%$ dos LSBs.

(b) $|M| = 5\%$ dos LSBs.

(c) $|M| = 10\%$ dos LSBs.

(d) $|M| = 25\%$ dos LSBs.

(e) $|M| = 50\%$ dos LSBs.

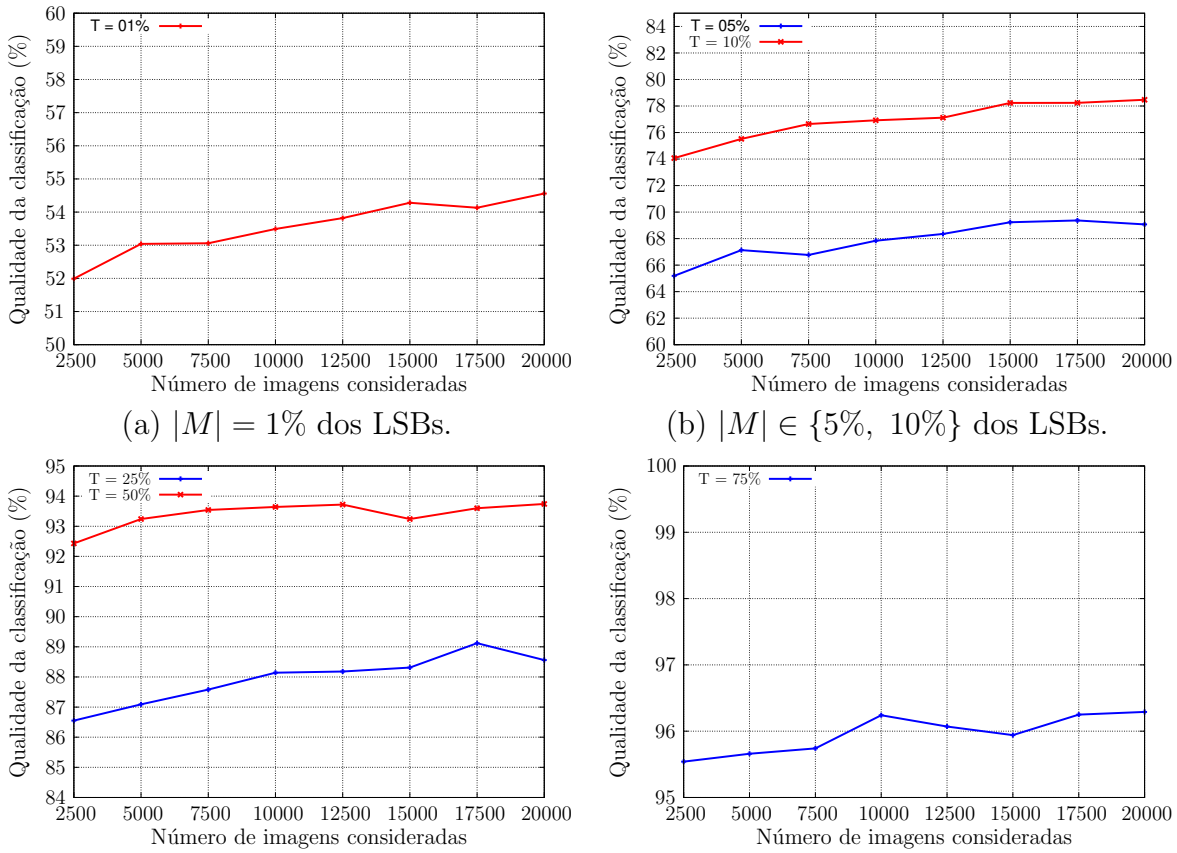
(f) $|M| = 75\%$ dos LSBs.

Figura 4.3: Número ideal de iterações para o classificador *Bagging* associado ao LDA.

4.3.3 Tamanho dos conjuntos de treinamento e de teste

Uma questão de bastante interesse relacionada à classificação é quando parar de treinar o classificador. Em nossos experimentos, nós verificamos que a cada 2.500 imagens adicionadas ao treinamento, nós obtemos uma melhoria de cerca de 1% na qualidade da classificação.

As Figuras 4.4(a-c) mostram os efeitos da utilização de mais imagens no conjunto de análise. Reportamos os resultados utilizando o classificador *Bagging* associado ao LDA. Utilizamos 37 iterações no processo de *Bagging*. Como utilizamos validação cruzada para reportar nossos resultados, quando analisamos 20.000 imagens, temos dez partições de 2.000 imagens cada. A cada passo da validação cruzada, procedemos o treinamento com nove partições e testamos em uma.



(a) $|M| = 1\%$ dos LSBs.

(b) $|M| \in \{5\%,\ 10\%\}$ dos LSBs.

(c) $|M| \in \{25\%,\ 50\%\}$ dos LSBs.

(d) $|M| = 75\%$ dos LSBs.

Figura 4.4: Tamanho ideal para o conjunto de treinamento utilizando *Bagging* associado ao LDA com 37 iterações. Não-estego *vs.* estego-imagens.

Neste trabalho, reportamos os resultados considerando as melhores condições de treinamento, isto é, quando temos um conjunto total de 20.000 imagens para analisar, embora o treinamento com este número de imagens seja um pouco mais demorado. É importante observarmos que o treinamento é feito apenas uma vez. A partir do momento em que conseguimos os coeficientes de classificação necessários ao funcionamento de um dado classificador, temos que o tempo de teste deste classificador é independente do número de imagens utilizadas na etapa de aprendizado.

4.3.4 Análise por classes

O contexto de uma imagem digital é muito variável. Temos que considerar os mais variados contextos possíveis na construção do banco de imagens. Existem imagens cujo contexto é complexo tais como imagens de obras de arte e paisagens (*outdoors*). Nestas imagens, o número de cores únicas[3] é mais próximo do número de *pixels* que em imagens mais simples e com menos detalhes como as imagens de ambientes internos (*indoors*). A detecção de conteúdo escondido em imagens de poucos detalhes é bem mais simples pois os artefatos inseridos nestas imagens pelo processo de mascaramento são mais óbvios que os artefatos inseridos em imagens com maior riqueza de detalhes. Apresentamos um exemplo de cada uma das classes analisadas nas Figuras 4.5(a-d).

As Figuras 4.6(a-b) mostram os impactos na qualidade de classificação quando consideramos alguns contextos diferentes no conjunto de imagens analisadas. Reportamos os resultados utilizando o classificador *Bagging* associado ao LDA. Utilizamos 37 iterações no processo de *Bagging*. Neste experimento, consideramos quatro classes de imagens:

1. **Obras de arte**. Composta por 2.780 imagens de alta resolução de obras de arte famosas.

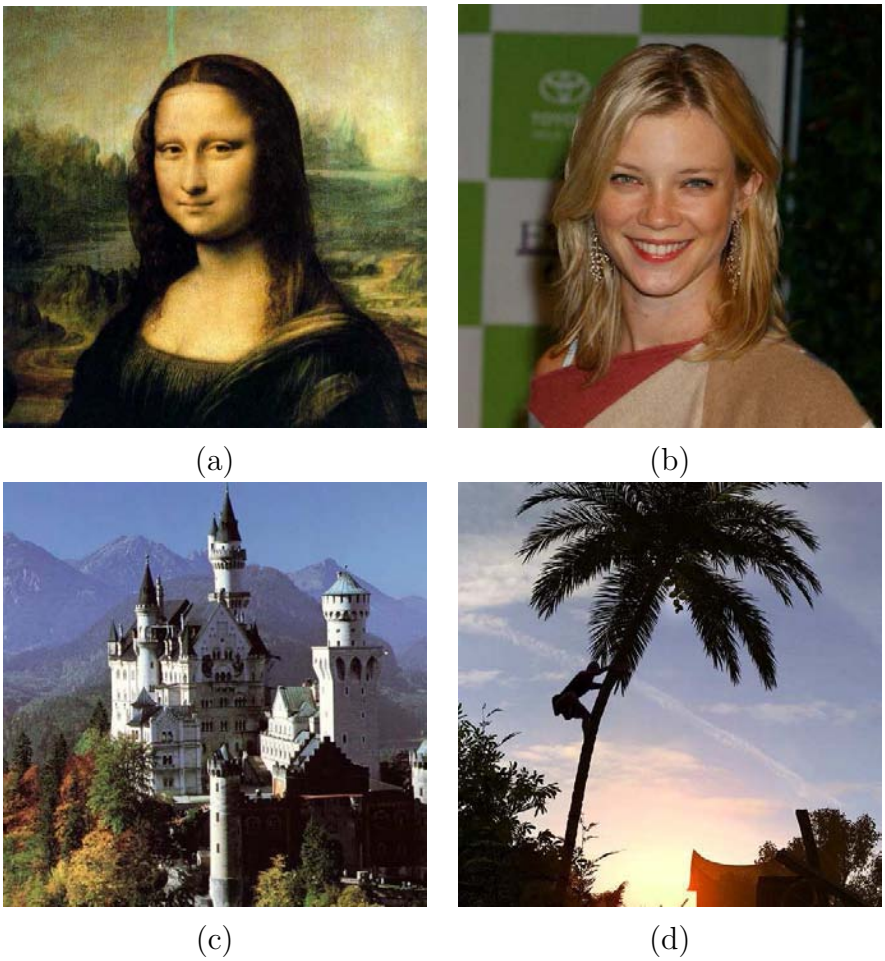
2. **Indoors**. Composta por 3.700 imagens retratando ambientes internos e de pouca variabilidade de contexto e de detalhes. Inclui fotos de pessoas em ambientes internos.

3. **Outdoors**. Composta por 3.400 imagens retratando paisagens e os mais variados ambientes externos. Estas imagens possuem grande variabilidade de contexto e de detalhes.

4. **CGI**. Composta por 470 imagens geradas em computador.

As quatro classes consideradas totalizam 10.350 imagens. Em cada etapa de análise, criamos um conjunto de treinamento com cerca de 7.000 imagens selecionadas aleatoriamente de três classes. Criamos o conjunto de teste utilizando todas as imagens da classe

[3]Cada cor diferente presente na imagem.



(a) (b)

(c) (d)

Figura 4.5: Um exemplo de cada uma das quatro classes analisadas. (a) Artes. (b) Indoors. (c) Outdoors. (d) CGI.

restante. O processo é repetido para cada combinação de classes possível. As classes mostradas na Figura 4.6 representam as classes de teste.

Os valores esperados reportados são referentes a uma seleção aleatória de 7.000 imagens considerando as quatro classes. As imagens remanescentes são utilizadas no teste. Olhando os resultados, concluímos que a classe *Artes* seguida pela classe *Outdoors* constituem as classes mais difíceis de detectarmos mensagens escondidas. Por outro lado, é importante notarmos o comportamento curioso relacionado à classe de imagens *Indoors*. Apesar de não possuirmos exemplares desta classe em nosso treinamento referente a este teste, os resultados obtidos são melhores que os valores esperados. Isto se deve ao fato de que esta classe de imagens possui contexto bem mais simples para detecção que as outras classes consideradas. Uma vez que tenhamos treinado o sistema para proceder a

classificação considerando classes mais difíceis, temos o comportamento necessário para classificar uma classe de imagens de contexto mais simples.

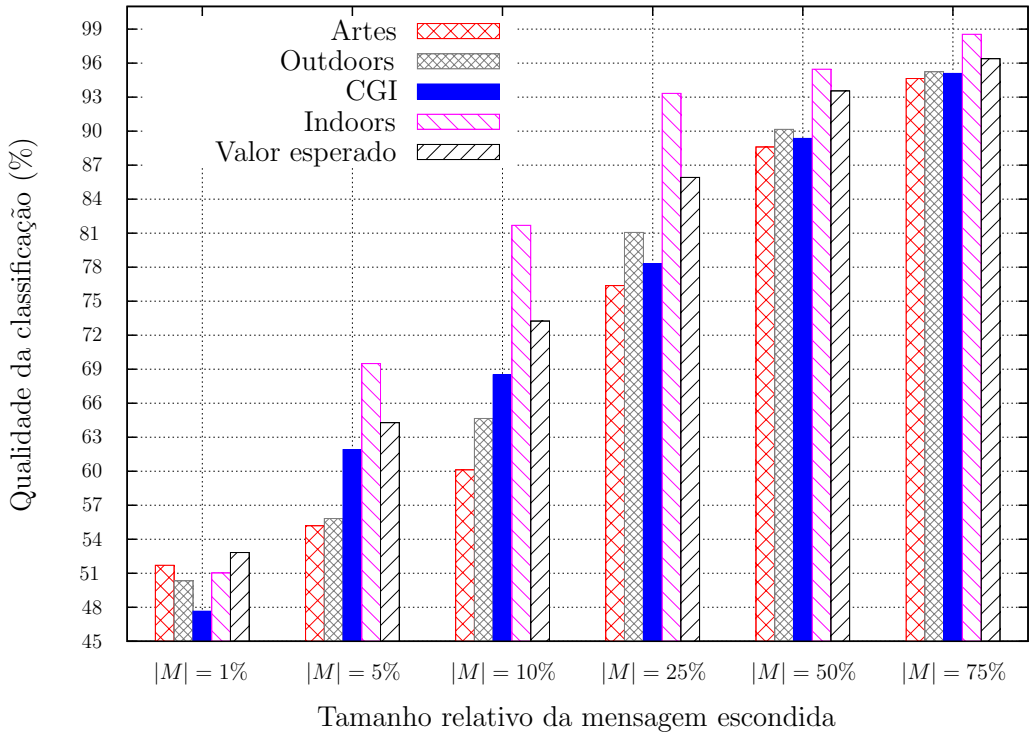


Figura 4.6: Análise por classes utilizando *Bagging* associado ao LDA com 37 iterações. Não-estego *vs.* estego-imagens. $|M| \in \{1\%,\ 5\%,\ 10\%, 25\%, 50\%, 75\%\}$ dos LSBs.

4.3.5 Considerações finais

Analisando nossos resultados (Tabelas 4.1 e 4.2), concluímos que quanto menor a mensagem escondida, pior é a performance da classificação. Quando analisamos imagens cujo conteúdo escondido possui tamanho relativo em torno de 1% dos LSBs disponíveis, a qualidade de detecção é pouco melhor que um jogo de adivinhação. Na prática, quando pedófilos utilizam esteganografia em imagens para vender suas imagens de pornografia infantil na *internet*, eles geralmente utilizam uma porção razoável da capacidade do canal LSB.

Uma imagem de cobertura típica de 24 *bits* e resolução 1024×768 *pixels* tem aproxidamente 294 KB de espaço disponível no canal LSB. Uma imagem JPEG [25] típica de resolução 800×600 *pixels* tem um tamanho aproximado de 75 KB. No caso em que

um pedófilo queira distribuir tal imagem escondida em uma imagem de cobertura típica, ele irá utilizar cerca 25% da capacidade do canal LSB. Nesta classe de problema, nossa abordagem detecta tais atividades com uma precisão de $\mu = 89,3\%$ e $\sigma = 0,6\%$ considerando o classificador SVM-RBF e a abordagem de Harris e Quads como regiões características.

4.4 A abordagem de Westfeld & Pfitzmann

Westfeld & Pfitzmann [54] desenvolveram uma abordagem de detecção que apenas detecta mensagens seqüencialmente escondidas a partir do primeiro LSB disponível na imagem de cobertura. Esta abordagem não é muito robusta à variabilidade de contextos das imagens. Além disso, esta abordagem não é robusta para detectar mensagens alteradas a partir de algum procedimento de mascaramento de mensagens que mantenha alguns perfis estatísticos básicos tais como média, variância e desvio padrão da imagem de cobertura [41].

Nossa abordagem de detecção trata estes problemas e melhora a qualidade da classificação em torno de 10 pontos percentuais em alguns casos. Infelizmente, os autores reportaram resultados para um número muito pequeno de imagens. Para comparar a abordagem de Westfeld & Pfitzmann com a Randomização Progressiva, nós alteramos a abordagem inicial dos autores de modo que tal abordagem seja capaz de detectar mensagens aleatoriamente distribuídas pela imagem e utilize um estágio de treinamento com as mesmas regiões características consideradas em nossa abordagem. A Tabela 4.3 e as Figuras 4.7(a-f) apresentam uma comparação entre as duas abordagens. Mostramos em fundo branco a abordagem de detecção de Westfeld & Pfitzmann e, em fundo cinza, a abordagem da Randomização Progressiva.

Os melhores resultados para a abordagem de Westfeld & Pfitzmann (WP) são obtidos utilizando o classificador LDA. Para mensagens de tamanho relativo de 5% dos LSBs disponíveis, nossa abordagem de Randomização Progressiva é superior à abordagem WP em aproximadamente cinco desvios padrões considerando o classificador LDA e em cerca de 18 desvios considerando o classificador SVM. Para mensagens de tamanho relativo de 25% dos LSBs disponíveis, nossa abordagem é superior à WP em aproximadamente cinco desvios padrões. Considerando o classificador LDA, para mensagens cujo tamanho relativo é superior a 50% dos LSBs disponíveis, a abordagem de WP é superior à Randomização Progressiva. No entanto, ao utilizarmos o classificador SVM, nossos resultados são superiores mesmo nestes casos. O classificador SVM não tem uma boa performance na abordagem de Westfeld & Pfitzmann.

	LDA		SVM-RBF		Tipo
	μ	σ	μ	σ	
01%	52,6%	0,9%	52,6%	0,1%	WP
	52,9%	1,6%	54,1%	0,9%	PR
05%	60,4%	0,8%	52,6%	0,1%	WP
	65,2%	1,0%	70,7%	0,9%	PR
10%	68,6%	0,9%	54,6%	4,1%	WP
	75,5%	0,7%	80,2%	0,5%	PR
25%	81,6%	0,6%	72,9%	1,9%	WP
	85,6%	0,8%	89,3%	0,6%	PR
50%	90,7%	0,5%	83,0%	0,6%	WP
	89,0%	0,6%	94,0%	0,5%	PR
75%	95,0%	0,4%	84,8%	0,9%	WP
	92,0%	0,6%	96,3%	0,3%	PR

Tabela 4.3: Abordagem de detecção de Westfeld & Pfitzmann (WP) *vs.* Randomização Progressiva (PR). μ and σ são referentes à validação cruzada.

4.5 A abordagem de Lyu & Farid

Lyu & Farid [12, 13, 31, 12] desenvolveram uma técnica que decompõe uma imagem em filtros de quadratura em espelho (QMFs – *Quadrature Mirror Filters*) [50]. Esta decomposição divide a imagem no domínio da freqüência em múltiplas escalas e orientações.

Os autores configuraram os parâmetros de seus classificadores para fixar a taxa de falsos positivos em 1%. Eles preferiram perder algumas imagens com conteúdo escondido a classificar erroneamente uma imagem sem conteúdo escondido.

A Tabela 4.4 e as Figuras 4.8(a-f) apresentam uma comparação entre as duas abordagens. Mostramos em fundo branco a abordagem de detecção de Lyu & Farid e, em fundo cinza, a abordagem da Randomização Progressiva.

Aqui nós computamos a qualidade da classificação como sendo a razão entre o número de estego-imagens corretamente classificadas e o número total de estego-imagens. Neste caso, não faz sentido considerarmos na qualidade de classificação as imagens não-estego uma vez que configuramos os parâmetros dos classificadores para termos uma taxa de falsos positivos em torno de 1%.

Analisando a Tabela 4.4, podemos concluir que nossos resultados (fundo cinza) têm maior exatidão que os resultados de Lyu & Farid. Nossa abordagem de Randomização Progressiva detecta mensagens com pequeno tamanho relativo (e.g. $|M| = 1\%$) com uma exatidão de aproximadamente dois pontos percentuais (cerca de dois desvios padrões)

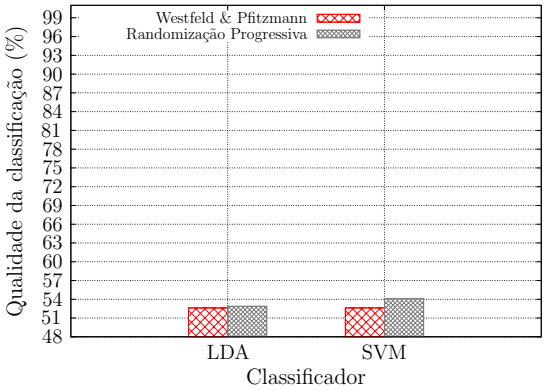
	LDA		**SVM-RBF**		**Tipo**
	μ	σ	μ	σ	
01%	1,3%	–	1,9%	–	LF
	3,2%	0,5%	3,6%	1,0%	PR
10%	2,8%	–	6,2%	–	LF
	7,0%	0,8%	15,8%	1,1%	PR
50%	16,8%	–	44,7%	–	LF
	24,2%	1,5%	53,1%	1,6%	PR
99%	42,3%	–	78,0%	–	LF
	95,8%	0,5%	97,0%	0,6%	PR

Tabela 4.4: Abordagem de detecção de Lyu & Farid (LF) *vs.* Randomização Progressiva (PR) considerando FPR = 1%. μ and σ são referentes à validação cruzada. Resultados de Lyu & Farid extraídos de [31, 13].

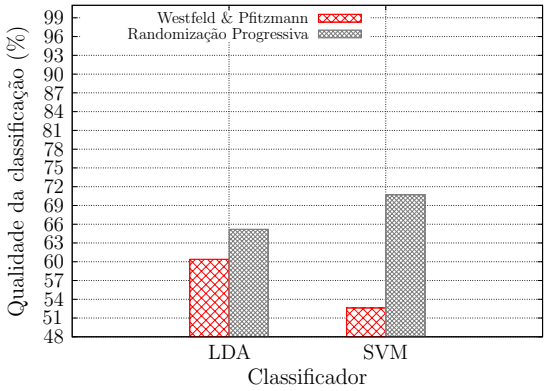
melhor que a abordagem de Lyu & Farid considerando os classificadores LDA e SVM. Para mensagens com médio tamanho relativo (e.g. $|M| = 50\%$), nossa abordagem apresenta uma melhoria de cerca de oito pontos percentuais (cerca de cinco desvios padrões) em relação à abordagem LF em ambos os classificadores. Quando consideramos mensagens que ocupam todo o espaço disponível no canal LSB (e.g. $|M| = 99\%$), nossa abordagem é 53 pontos percentuais mais exata que a abordagem LF para o classificador LDA e aproximadamente 19 pontos percentuais (cerca de 31 desvios padrões) mais exata para o classificador SVM.

Quando fixamos a taxa de falsos positivos (FPR) em 1%, a abordagem utilizando SVM-RBF, embora computacionalmente mais cara e de implementação mais complexa, produz melhores resultados que simplesmente utilizarmos um discriminante linear (LDA). Em ambas abordagens, no entanto, quanto menor o tamanho relativo da mensagem escondida, pior é a performance da classificação.

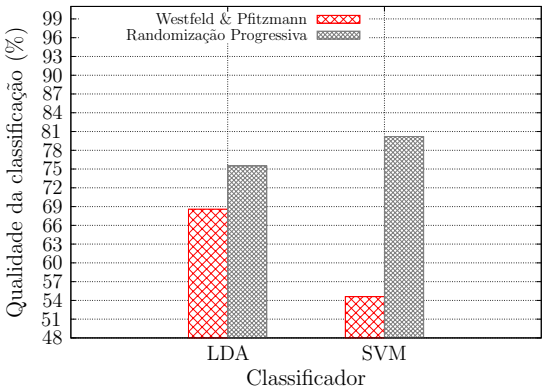
Nossa abordagem de detecção ainda não considera as vantagens da coerência espacial nas imagens. No entanto, nossa abordagem apresenta melhores resultados que as técnicas comparáveis existentes, inclusive em relação àquelas que utilizam coerência espacial nas imagens tais como [12, 13, 31, 12].



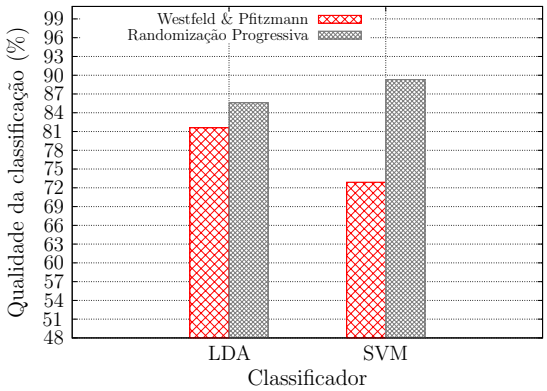
(a) $|M| = 1\%$ dos LSBs.



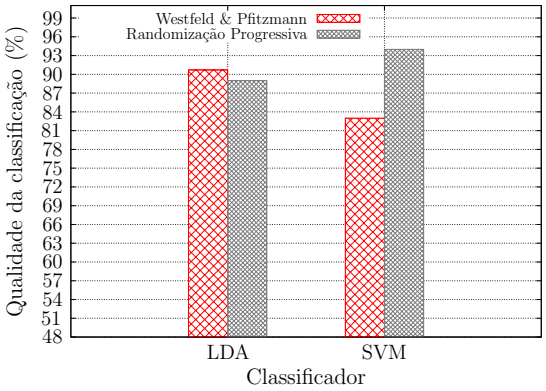
(b) $|M| = 5\%$ dos LSBs.



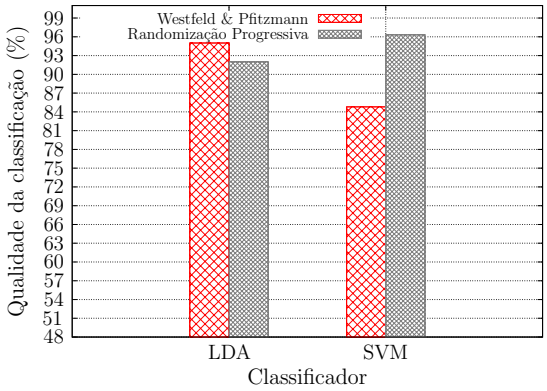
(c) $|M| = 10\%$ dos LSBs.



(d) $|M| = 25\%$ dos LSBs.

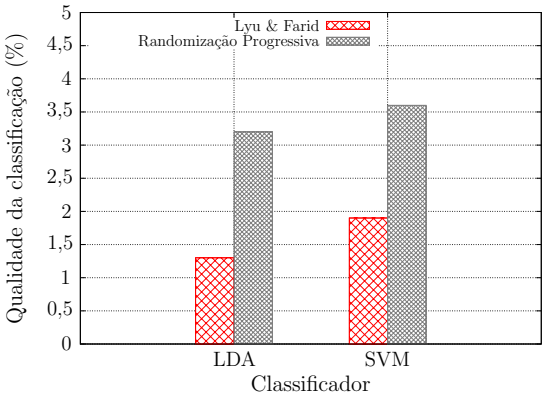


(e) $|M| = 50\%$ dos LSBs.

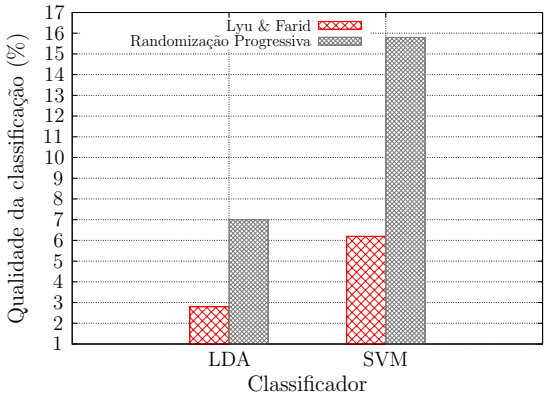


(f) $|M| = 75\%$ dos LSBs.

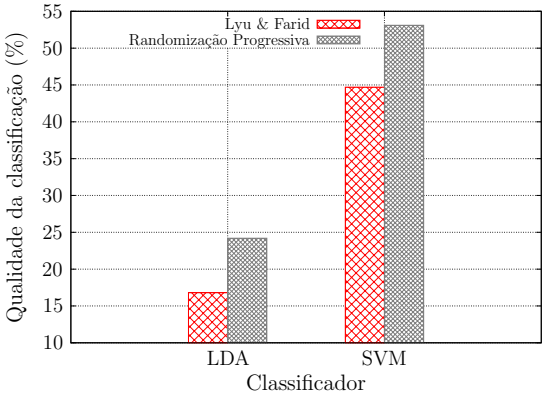
Figura 4.7: Abordagem de detecção de Westfeld & Pfitzmann *vs.* Randomização Progressiva.



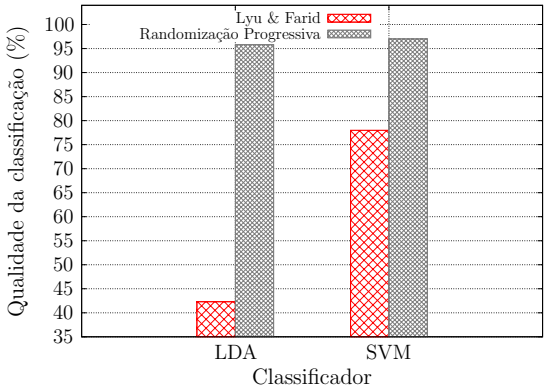
(a) $|M| = 1\%$ dos LSBs.



(b) $|M| = 10\%$ dos LSBs.



(c) $|M| = 50\%$ dos LSBs.



(d) $|M| = 99\%$ dos LSBs.

Figura 4.8: Abordagem de detecção de Lyu & Farid *vs.* Randomização Progressiva.

Capítulo 5

Conclusões e trabalhos futuros

Se digo que sei, eu paro de pensar.
Se digo que não sei, eu continuo pensando
e acabo compreendendo.
(Albert Einstein)

Neste trabalho, nós apresentamos um novo conjunto de descritores estatísticos de imagem que nos permitem detectar mensagens aleatoriamente escondidas no canal LSB de imagens digitais.

Nossa abordagem de Randomização Progressiva utiliza apenas descritores estatísticos dos LSBs e apresenta uma melhor qualidade de classificação que os métodos comparáveis descritos na literatura [23, 54, 18, 17, 31, 13]. Isto indica que nosso método constitui uma abordagem efetiva de detecção de mensagens escondidas em imagens digitais.

A detecção de mensagens de tamanho relativo muito pequeno constitui um problema ainda em aberto. Quanto menor o tamanho da mensagem escondida, pior a performance da classificação.

Embora a detecção de mensagens de tamanho relativo muito pequeno (e.g. $|M| = 1\%$ da capacidade do canal LSB) seja praticamente um jogo de adivinhação, em situações práticas, tais como quando pedófilos utilizam imagens para vender suas imagens de pornografia infantil, eles usam uma porção razoável do canal LSB (e.g. $|M| = 25\%$) conforme discutimos na Seção 4.3.5. Nesta classe de problema, nossa abordagem detecta tais atividades com uma precisão de $\mu = 89,3\%$ e $\sigma = 0,6\%$ considerando o classificador SVM-RBF e a abordagem de Harris e Quads como regiões características.

Nossos resultados apontam a abordagem Bagging associada ao LDA como a que nos proporciona o melhor custo/benefício considerando tempo de classificação e complexidade de implementação. Segundo os mesmos requisitos, escolhemos 37 iterações como o melhor custo/benefício para a abordagem *Bagging*. O classificador SVM-RBF não é beneficiado

pela abordagem *Bagging* já que utiliza no cálculo no procedimento de classificação apenas os elementos próximos à fronteira de decisão (margem). Isto também foi verificado experimentalmente.

Mostramos que a utilização de quatro regiões de Harris e quatro regiões de Quads tem uma qualidade de classificação inferior do que quando utilizamos oito regiões constantes igualmente distribuídas pela imagem. No entanto, a utilização das regiões de Harris é uma tentativa de achar mascaramentos localizados que um indivíduo pode fazer em áreas de grandes detalhes em uma imagem e, portanto, deve ser considerada nesses casos. Observamos também que quanto maior a mensagem escondida, menor é a diferença nas qualidades de classificação de ambos os métodos.

Em nossos experimentos, verificamos que a cada 2.500 imagens adicionadas ao treinamento, obtemos uma melhoria de cerca de 1% na qualidade da classificação. Verificamos também que o contexto de uma imagem digital é muito variável e temos que considerar os mais variados contextos possíveis na construção do banco de imagens. A detecção de conteúdo escondido em imagens de poucos detalhes é bem mais simples pois os artefatos inseridos nestas imagens pelo processo de mascaramento são mais óbvios que os artefatos inseridos em imagens com maior riqueza de detalhes. Concluímos que a classe *Artes* seguida pela classe *Outdoors* constituem as classes mais difíceis de detectarmos mensagens escondidas. Por outro lado, a classe *Indoors* constitui a classe mais fácil para detecção.

Nossos trabalhos futuros incluem uma análise teórica da relação entre a Randomização Progressiva, entropia e a teoria da informação. Queremos obter limites assintóticos, provas de corretude e limitações de nosso método. Também estamos interessados na consideração de outros descritores estatísticos bem como em uma análise multi-escala de nossa abordagem de detecção para levarmos em conta as vantagens da coerência espacial.

Finalmente, nós planejamos aplicar nossa técnica para a detecção de diferentes tipos de métodos de esteganografia, e para a resolução de outros problemas de classificação de imagens.

Apêndice A

Técnicas de esteganálise

A.1 Análise RS

Apresentada por Jessica Fridrich [16], esta técnica consiste na análise das inter-relações entre os planos de cores presente nas imagens analisadas. A classificação é feita pontualmente, sem utilização de treinamento.

Algumas extensões do modelo inicial foram feitas e em algumas delas é possível também estimar o tamanho da mensagem escondida [16]. O método consiste em analisar a capacidade de mascaramento sem perdas do plano LSB[1].

Em um grande número de imagens, o plano LSB é essencialmente disperso, muito próximo de um estado pseudo-aleatório, e não possui qualquer artefato reconhecível. A análise da capacidade de mascaramento sem perdas reflete o fato de que mesmo que o plano LSB pareça ser independente dos demais planos, de alguma forma existe relações entre os mesmos. Esta relação é não linear e pode ser estimada pela capacidade de mascaramento sem perdas. Esta estimativa é feita através da simulação artificial de um novo processo de mascaramento em uma imagem que precisa ser classificada como tendo ou não uma mensagem escondida. Esta simulação consiste na criação de funções que simulam o processo de mascaramento e também na divisão da imagem analisada em grupos.

Considere a imagem testada I com $W \times H$ *pixels*. Cada *pixel* tem valores dados por um conjunto P. Para uma imagem de 8 bpp (*bits por pixel*), temos $P = \{0, \ldots, 255\}$.

Dividimos I em grupos G de *pixels* disjuntos de n *pixels* adjacentes. Como exemplo, podemos escolher grupos de $n = 4$ *pixels* adjacentes. Feito isso, definimos uma função de discriminação f responsável por atribuir um número real $f(x_1, \ldots, x_n) \in \Re$ para cada

[1]Mascaramento sem perdas (*lossless data embedding*) é o mascaramento em que todos os *bits* escondidos podem ser recuperados. Diferencia-se do mascaramento com perdas (*lossy*) em que é recuperada apenas uma estimativa dos *bits* da mensagem.

grupo de *pixels* $G = (x_1, \ldots, x_n)$. O objetivo de f é capturar a regularidade (suavidade) do grupo de *pixels* G. Quanto mais distinto for o grupo de *pixels*, maior o valor da função de discriminação. Um exemplo de função de discriminação pode ser

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|. \tag{A.1}$$

Finalmente, definimos uma operação inversível F sobre P chamada *flipping*. Por *flipping* entendemos a permutação dos níveis de cores que consiste em dois ciclos. Assim, F tem a propriedade que $F^2 =$ Identidade ou $F(F(x))) = x$ para todo *pixel* $x \in P$. A função $F_1 : 0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255$ corresponde a inverter o LSB de cada nível de cor.

Adicionalmente, podemos definir uma função $F_{-1} : -1 \leftrightarrow 0, 1 \leftrightarrow 2, \ldots, 255 \leftrightarrow 256$ de deslocamento (*shifting*). No caso de *inversão* e *deslocamento*,

$$F_{-1}(x) = F_1(x + 1) - 1 \text{ para todo } x \in P. \tag{A.2}$$

Podemos calcular F_{-1} a partir da função de inversão F_1. Para completar, definimos F_0 como sendo a função de identidade $F_0(x) = x$ para todo $x \in P$.

Para aplicarmos diferentes funções em diferentes *pixels*, devemos usar uma máscara $\mathcal{M}$ que irá denotar quais os *pixels* deverão sofrer alterações. A máscara $\mathcal{M}$ é uma n-tupla com valores $\{-1, 0, 1\}$. O valor -1 denota a aplicação da função F_{-1}, 1 denota a aplicação da função F_1 e 0 denota a aplicação da função F_0 (identidade) sobre os *pixels* do grupo analisado. Definimos a máscara negativa $-\mathcal{M}$ como o complemento de $\mathcal{M}$.

Aplicamos a função de discriminação f, bem como as funções F_1, F_{-1} e F_0 definidas mediante uma máscara $\mathcal{M}$ sobre os grupos G para classificá-los em três categorias diferentes: $R_\mathcal{M}$, $S_\mathcal{M}$ e $U_\mathcal{M}$:

- Grupos regulares : $G \in R_\mathcal{M} \Leftrightarrow f(F_\mathcal{M}(G)) > f(G)$
- Grupos singulares : $G \in S_\mathcal{M} \Leftrightarrow f(F_\mathcal{M}(G)) < f(G)$
- Grupos não-usáveis : $G \in U_\mathcal{M} \Leftrightarrow f(F_\mathcal{M}(G)) = f(G)$

Da mesma forma, classificamos os grupos $R_{-\mathcal{M}}$, $S_{-\mathcal{M}}$ e $U_{-\mathcal{M}}$ sob a máscara $-\mathcal{M}$. O objetivo da função $F_\mathcal{M}$ é perturbar os *pixels* de uma forma pouco significativa tal como no processo de mascaramento de uma mensagem.

De forma geral, temos que

$$\frac{|R_\mathcal{M}| + |S_\mathcal{M}|}{T} \leq 1 \text{ e } \frac{|R_{-\mathcal{M}}| + |S_{-\mathcal{M}}|}{T} \leq 1, \tag{A.3}$$

onde T representa o número total de grupos G criados.

A hipótese estatística para o método é que, em imagens típicas, o valor esperado de $R_{\mathcal{M}}$ é aproximadamente igual ao de $R_{-\mathcal{M}}$ e o mesmo é verdade para $S_{\mathcal{M}}$ e $S_{-\mathcal{M}}$

$$R_{\mathcal{M}} \cong R_{-\mathcal{M}} \text{ e } S_{\mathcal{M}} \cong S_{-\mathcal{M}}. \tag{A.4}$$

A randomização do plano LSB força a diferença entre $R_{\mathcal{M}}$ e $S_{\mathcal{M}}$ para zero à medida que o tamanho p da mensagem escondida cresce. Depois de alterar os LSBs de 50% dos *pixels* (é o que acontece quando escondemos uma mensagem aleatoriamente distribuída em todos os *pixels*), obtemos $R_{\mathcal{M}} \cong S_{\mathcal{M}}$, isto é o mesmo que dizer que a capacidade de mascaramento no plano LSB agora é zero. O fato surpreendente é que um efeito contrário acontece com $R_{-\mathcal{M}}$ e $S_{-\mathcal{M}}$, sua diferença aumenta proporcionalmente ao tamanho da mensagem escondida. A estimativa da presença ou ausência de uma mensagem escondida bem como do seu possível tamanho é feita através do cálculo por extrapolação da interseção dos pontos $R_{\mathcal{M}}(50)$ e $S_{\mathcal{M}}(50)$ que aparecem na Figura A.1. No entanto, a Equação A.4 é apenas uma suposição sendo dependente do contexto da imagem analisada.

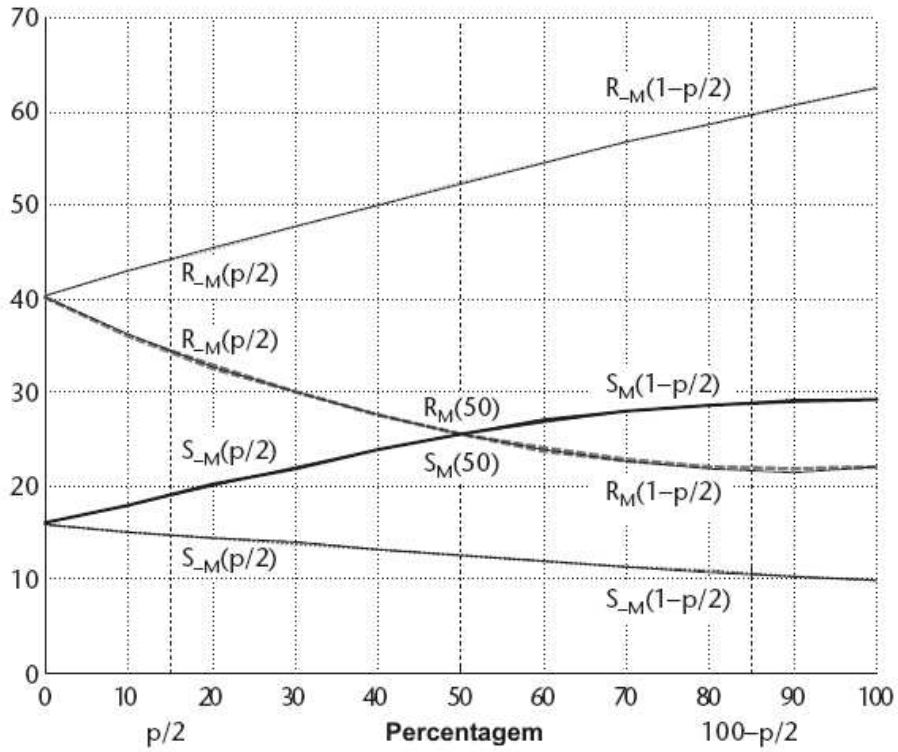


Figura A.1: Diagrama RS de uma imagem. O eixo x é a percentagem de *pixels* cujos LSBs foram invertidos pela função $F_{\mathcal{M}}$. O eixo y é o número relativo de grupos regulares e singulares sob as máscaras $\mathcal{M} = [0, 1, 1, 0]$ e $-\mathcal{M} = [0, -1, -1, 0]$.

A.2 Análise de cores únicas no cubo RGB

Proposto por Jiri Fridrich et al. [18], este método é baseado na análise estatística de cores do cubo RGB da imagem considerada. As técnicas de mascaramento LSB consideram que o plano menos significativo de *bits* é essencialmente randômico. A substituição dos *bits* deste plano por uma mensagem também randômica não irá inserir artefatos detectáveis. No entanto, isso é essencialmente verdade apenas se o número de cores únicas[2] presente na imagem for comparável ao número de *pixels* da mesma. Entretanto, sabemos que o número de cores únicas em imagens de cores verdadeiras (*true colors*) é tipicamente menor. A razão entre o número de cores únicas e o número de *pixels* variam de 1:2 para imagens de alta qualidade a até 1:6 para imagens de menor resolução [18]. Isto significa que muitas imagens de cores verdadeiras possuem uma paleta de cores relativamente pequena e, após o mascaramento LSB, a nova paleta de cores conterá muitos pares de cores relativamente muito próximas.

Considere U o número de cores únicas em uma imagem. Olhando apenas em U, seja P o número de pares de cores próximas na paleta da imagem. Duas cores (R_1, G_1, B_1) e (R_2, G_2, B_2) são próximas se

$$(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2 \leq 3. \tag{A.5}$$

O número de todos os pares de cores é dado pelo binômio

$$\binom{U}{2} \geq P. \tag{A.6}$$

A razão R entre o número de pares de cores próximas e o número de todos os pares de cores

$$R = \frac{P}{\binom{U}{2}}, \tag{A.7}$$

indica o número relativo de cores próximas na imagem. Após o mascaramento, o número de cores únicas na imagem irá aumentar. Para uma imagem que não contenha uma mensagem escondida, o número de pares de cores próximas em relação ao número de pares de todas as cores possíveis será menor que para uma imagem que contenha uma mensagem escondida. Embora esta afirmação possa ser verificada empiricamente, encontrar um limiar para R que separe imagens normais de imagens que contenham mensagens escondidas é uma tarefa quase impossível dado que o número de cores únicas em U pode variar muito. No entanto, em uma imagem I que já contenha um conteúdo escondido, o mascaramento de uma mensagem adicional em I não altera significativamente o valor de R. Por outro lado, se I não contém uma mensagem escondida, o valor de R cresce

[2]Cada cor diferente presente na imagem.

significativamente após este procedimento. A razão R então é utilizada como critério de decisão para classificar uma imagem como tendo ou não uma mensagem escondida. O processo de decisão, sobre a imagem I de tamanho $W \times H$ testada, consiste em:

1. calcular a razão R entre o número de todos os pares possíveis de cores próximas P e o número de todas cores possíveis U

$$R = \frac{P}{\binom{U}{2}}; \tag{A.8}$$

2. usando o mascaramento LSB, mascare uma mensagem de tamanho $3\alpha WH$ pseudo-aleatoriamente em I onde α é discutido posteriormente.

3. Denote as quantidades correspondentes para a imagem I' gerada no passo anterior como U' e P' e calcule a razão R' para I' com a mensagem de teste

$$R' = \frac{P'}{\binom{U'}{2}}. \tag{A.9}$$

A hipótese estatística é que se I já contém uma mensagem escondida de tamanho significativo, as duas razões são praticamente as mesmas $R \cong R'$. Entretanto, se I não contém uma mensagem escondida, temos que $R' > R$. Para facilitar o cálculo da separabilidade, podemos definir a razão R'/R.

Caso o tamanho da mensagem escondida seja muito pequeno, as duas razões também serão muito próximas. Desta forma, a escolha correta de α deve ser feita de modo a minimizar o número de falsas detecções. O valor proposto por [18] e descoberto empiricamente é $\alpha = 5\%$. Infelizmente, a dependêndia de parâmetros *ad-hoc* e do contexto da imagem analisada enfraquecem o método.

A.3 Taxa de inversão da energia do gradiente

Apresentada por Li Shi et al. [47], esta abordagem consiste em analisar a variação da energia do gradiente (EG), devido ao processo de mascaramento, dos planos de *bits* das imagens analisadas. Para um maior entendimento, considere um sinal $I(n)$ unidimensional. O gradiente, $r(n)$, de um sinal antes do mascaramento é dado por

$$r(n) = I(n) - I(n-1). \tag{A.10}$$

A *energia do gradiente*, EG, de $I(n)$ é dada por

$$EG = \sum |I(n) - I(n-1)|^2 = \sum r(n)^2. \tag{A.11}$$

Depois do processo de mascaramento de um sinal $S(n)$ no sinal original, $I(n)$ torna-se $I'(n)$ e o gradiente torna-se

$$\begin{aligned} r(n) &= I(n) - I(n-1) \\ &= (I(n) + S(n)) - (I(n-1) + S(n-1)) \\ &= r(n) + S(n) - S(n-1). \end{aligned} \tag{A.12}$$

A função de distribuição de probabilidade de $S(n)$ é dada pela Equação A.13

$$\begin{cases} \rho(S(n)) \approx 0 &= \frac{1}{2} \\ \rho(S(n)) \approx \pm 1 &= \frac{1}{4} \end{cases} \tag{A.13}$$

Após o processo de mascaramento, a nova *energia do gradiente*, EG', é

$$\begin{aligned} GE' &= \sum |r(n)|^2 = \sum |r(n) + S(n) - S(n-1)|^2 \\ &= \sum |r(n) + \Delta(n)|^2, \text{onde } \Delta(n) = S(n) - S(n-1). \end{aligned} \tag{A.14}$$

Para proceder o processo de detecção, é necessário definirmos o processo de inversão dos *bits* do plano LSB da imagem. Para tal, criamos uma operação inversível F sobre um conjunto P de *bits* invertidos como já definido para a *Análise RS* (Seção A.1).

Caso a imagem de cobertura tenha $W \times H$ *pixels* e o tamanho da mensagem escondida seja $p \leq W \times H$, a operação F resulta em três propriedades:

1. Para $p = W \times H$, há $\dfrac{W \times H}{2}$ *pixels* com LSB invertido. Isto implica que a razão de mascaramento é de 50% e a *energia do gradiente* é dada por $EG = \left(\dfrac{W \times H}{2}\right)$.
2. A *energia do gradiente* da imagem original é dada por $EG(0)$. Ao fazer a inversão de todos os LSBs a partir da operação F, a *energia do gradiente* é $EG = W \times H$.
3. Para $p < W \times H$, há $\dfrac{p}{2}$ *pixels* com seu LSB invertido. Denotamos a imagem modificada como $I(\dfrac{p}{2})$. A *energia do gradiente* correspondente é dada por $EG = \dfrac{p/2}{W \times H} = EG(0) + p$. Caso a operação F seja aplicada sobre a imagem $I(\dfrac{p}{2})$, a *energia do gradiente* resultante é $EG = \dfrac{W \times H - p/2}{W \times H}$.

A partir do algoritmo de inversão proposto, o processo de detecção é como segue:

1. Calcule a *energia do gradiente* da imagem de teste $EG\left(\dfrac{p/2}{W \times H}\right)$;

2. Aplique a operação F sobre a imagem de teste e calcule $EG\left(\frac{W \times H - p/2}{W \times H}\right)$;
3. Calcule $EG\left(\frac{W \times H}{2}\right) = \left[EG\left(\frac{p/2}{W \times H}\right) + EG\left(\frac{W \times H - p/2}{W \times H}\right)\right]/2$;
4. $EG(0)$ é baseado em $EG\left(\frac{W \times H}{2}\right) = EG(0) + W \times H$;
5. Finalmente, o tamanho estimado para a mensagem escondida é dado pela expressão $p' = EG\left(\frac{p/2}{W \times H}\right) - EG(0)$.

A.4 Análise de estatísticas de alta ordem

Apresentada por Lyu e Farid [12, 31, 30, 13], esta abordagem de detecção consiste na construção de modelos estatísticos de alta ordem para imagens naturais e na busca por desvios nestes modelos.

As imagens naturais possuem regularidades que podem ser detectadas com estatísticas de alta ordem através de uma decomposição *wavelet*, por exemplo [30]. O processo de mascaramento de uma mensagem altera significativamente estas estatísticas tornando o processo de mascaramento matematicamente detectável.

Após a construção dos modelos de alta ordem, é necessário utilizarmos classificadores capazes de dizer se uma dada imagem possui ou não uma mensagem escondida.

O processo de decomposição das imagens usando funções base que são localizadas no domínio espacial de orientação e escala é extremamente útil em aplicações como compressão e codificação de imagens, remoção de ruído entre outras. Isto se deve ao fato destas decomposições exibirem regularidades estatísticas que podem ser exploradas.

Lyu e Farid aplicaram uma decomposição baseada nos *filtros de quadratura em espelho* (QMFs – *Quadrature Mirror Filters*) [50]. Esta decomposição divide a imagem no domínio da freqüência em múltiplas escalas e orientações. Esta decomposição é feita através da aplicação de filtros de passa-baixas e passa-altas sobre a imagem gerando quatro sub-bandas: *vertical, horizontal, diagonal* e de *passa-baixas*. Escalas subseqüentes são criadas aplicando-se o processo novamente sobre a sub-banda de *passa-baixas*. Denotamos as sub-bandas vertical, horizontal e diagonal em uma dada escala $\{i = 1 \ldots n\}$ por $V_i(x, y)$, $H_i(x, y)$, $D_i(x, y)$, respectivamente. A Figura A.2 demonstra o processo. A partir desta decomposição da imagem, criamos um modelo estatístico composto de *média*, *variância*, *assimetria* e *curtose* dos coeficientes da sub-banda em cada orientação e escala $\{i = 1 \ldots n\}$. Estas estatísticas caracterizam os *coeficientes básicos de distribuição*.

Para um maior grau de exatidão no processo de classificação, um segundo conjunto de estatísticas é necessário. Este segundo conjunto é baseado nos erros de um *preditor linear*

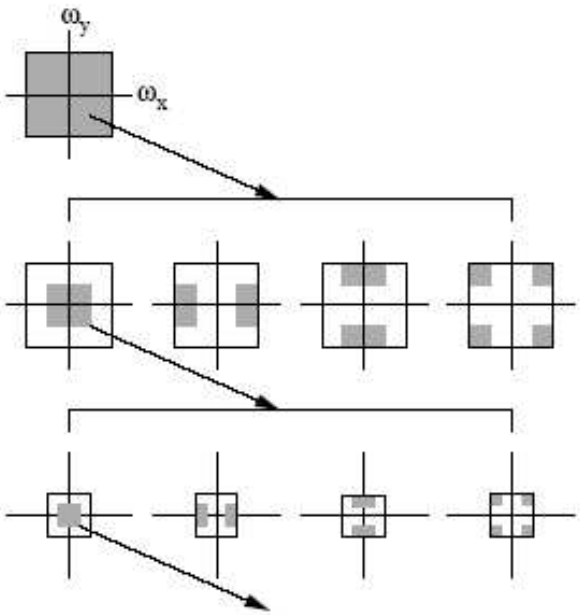


Figura A.2: Decomposição da imagem no domínio da freqüência em múltiplas escalas e orientações.

ótimo de coeficentes de magnitude. Os coeficientes das sub-bandas são correlacionados com seus vizinhos espaciais, de escala, e de orientação [5].

Para um maior entendimento, considere a sub-banda vertical, $V_i(x, y)$, na escala i. Um preditor linear para a magnitude destes coeficientes em um subconjunto de todos os possíveis vizinhos é dado por

$$\begin{aligned} V_i(x,y) &= w_1 V_i(x-1,y) + w_2 V_i(x+1,y) + w_3 V_i(x,y-1) \\ &\quad + w_4 V_i(x,y+1) + w_5 V_{i+1}(\frac{x}{2},\frac{y}{2}) + w_6 D_i(x,y) \\ &\quad + w_7 D_{i+1}(\frac{x}{2},\frac{y}{2}), \end{aligned} \tag{A.15}$$

onde w_k denota os valores escalares de peso dos coeficientes. Os coeficientes do erro são calculados através da minimização da *função de erro quadrática*

$$E(w) = [V - Qw]^2, \tag{A.16}$$

onde $w = (w_1, \ldots, w_7)^T$, V contém os coeficientes de magnitude de $V_i(x, y)$ dispostos em um vetor coluna e Q os coeficientes de magnitude dos vizinhos como especificado na Equação A.15. A função de erro é minimizada através da diferenciação com respeito a w

$$\frac{dE(w)}{dw} = 2Q^T[V - Qw]. \tag{A.17}$$

Após algumas simplificações, podemos calcular w_k diretamente através do *erro do log* no preditor linear

$$E = \log_2(V) - \log_2(|Qw|). \tag{A.18}$$

Todo o processo é recursivamente aplicado para cada sub-banda em cada escala e orientação. Ao final, temos um total de $12(n-1)$ estatísticas de erro mais $12(n-1)$ estatísticas básicas, totalizando $24(n-1)$ estatísticas para o vetor característico que deverá ser analisado pelo classificador escolhido. Lyu e Farid reportaram seus resultados utilizando os classificadores SVM, e LDA.

A.5 Métricas de qualidade de imagens

Estas métricas são utilizadas, de forma geral, na avaliação de codificação de artefatos, predição de performance de algoritmos de *Visão Computacional*, perda de qualidade devido a inadequabilidade de algum sensor, entre outras aplicações.

Nesta abordagem, proposta por Ismail Avcibas et al. [1, 2, 3], essas mesmas métricas são utilizadas para construir um discriminador de imagens de cobertura (naturais) de estego-imagens (com conteúdo escondido) através da utilização de *regressão multivariada*.

Dado que o mascaramento de uma mensagem pode ser interpretado como um sinal w adicionado ao sinal da imagem de cobertura f, temos que a estego-imagem gerada após o processo de mascaramento é dada por $g = f + w$. O discriminador é treinado sobre um conjunto de imagens de cobertura e de estego-imagens de modo a conseguir os coeficientes de qualidade de imagem que sejam capazes de separar as duas classes de imagens. Dado que no processo de *esteganálise* a imagem de cobertura quase sempre não está disponível para análise, é feita uma estimativa da imagem de cobertura através de filtragem baseada no filtro de passa-baixas Gaussiano [20]. No entanto, esta estimativa é extremamente dependente do contexto da imagem analisada.

As principais métricas de qualidade utilizadas são média angular, distância de fase de bloco espectral, distância espectral ponderada da mediana de bloco, erro médio quadrático normalizado do Sistema Visual Humano (SVH), medida de correlação de Czekznowski e erro médio absoluto.

Após os cálculos de cada coeficiente para todas as imagens do conjunto de treinamento, os autores propõem a regressão normalizada aos valores -1 e 1 para cada coeficiente. No modelo de regressão, cada decisão é expressa por y_i num conjunto de n imagens de observação.

Uma função linear dos coeficientes de qualidade de imagem é dada por

$$\begin{cases} y_1 &= \beta_1 x_{11} + \beta_2 x_{12} + \ldots + \beta_q x_{1q} + \epsilon_1 \\ y_2 &= \beta_2 x_{21} + \beta_2 x_{22} + \ldots + \beta_q x_{2q} + \epsilon_2 \\ &\vdots \\ y_N &= \beta_N x_{N1} + \beta_2 x_{12} + \ldots + \beta_q x_{Nq} + \epsilon_N. \end{cases} \tag{A.19}$$

Nesta equação, x_{ij} indica os coeficientes de qualidade onde o primeiro índice i indica a i-imagem, $i = 1, \ldots, N$, e o segundo indica a métrica de qualidade considerada, $j = 1, \ldots, q$ e q é o número total de métricas sob consideração. Os coeficientes de regressão são expressos por β_k e o erro aleatório[3] adicionado é expresso por ϵ.

Uma vez que estes coeficientes sejam obtidos na etapa de treinamento, eles podem ser utilizados na etapa de teste. Dado uma imagem na etapa de teste, primeiro é obtido uma versão filtrada desta numa tentativa de estimar a imagem original de cobertura. Utilizando os coeficientes de predição, é feita a regressão até um valor de saída ser obtido. Caso o valor de saída supere o limiar 0 então a decisão sobre a hipótese estatística é que a imagem possui uma mensagem escondida. Caso contrário, a decisão é que a imagem não contém uma mensagem escondida.

A.6 Métricas de tons contínuos e pares de amostragem

Proposta por Sorina Dumitrescu et al. [11, 10], esta abordagem consiste em analisar as relações de identidade estatística existentes sobre alguns conjuntos de *pixels* considerados. As identidades observadas são muito sensíveis ao mascaramento LSB e as mudanças nestas identidades podem indicar a presença de conteúdo escondido.

Para um maior entendimento desta abordagem, considere um particionamento de uma imagem dentro de pares de *pixels* (u, v) horizontalmente adjacentes. Seja $\mathcal{P}$ o conjunto de todos estes pares. Defina os subconjuntos X e Y de $\mathcal{P}$:

- X é o conjunto de pares $(u, v) \in \mathcal{P}$ tais que v é par e $u < v$, ou v é ímpar e $u > v$.
- Y é o conjunto de pares $(u, v) \in \mathcal{P}$ tal que v é par e $u > v$, ou v é ímpar e $u < v$.

X e Y são importantes para a *esteganálise* porque

$$|X| = |Y|. \tag{A.20}$$

[3]Este erro denota a variação natural que pode haver entre as amostras consideradas.

A relação exposta pela Equação A.20 é verdadeira para imagens sem conteúdo escondido. Defina Z como o subconjunto de pares $(u, v) \in \mathcal{P}$ tal que $u = v$. Além disso, considere a partição do subconjunto Y em dois subconjuntos: W e V, com W sendo o conjunto de pares em $\mathcal{P}$ da forma $(2k, 2k+1)$ ou $(2k+1, 2k)$ e $V = Y - W$. Neste caso, k demonstra a seleção de um par de *pixels* cuja diferença $|u-v| = 1$. Os conjuntos X, V, W e Z são chamados conjuntos primários, logo $\mathcal{P} = X \cup W \cup V \cup Z$.

Quando uma mensagem é escondida no plano LSB, o mascaramento modifica o valor de alguns *pixels* ao inverter seus LSBs. Assim, a cardinalidade dos *pixels* pertencentes a $\mathcal{P}$ irá mudar. Há quatro casos possíveis:

1. ambos os valores u e v não são modificados;
2. apenas u é modificado;
3. apenas v é modificado;
4. ambos os valores u e v são modificados.

Caso esteja ocorrendo a situação 1 (2,3,4) dizemos que os padrões de modificação devido ao mascaramento LSB é 00 (10, 01, 11, respectivamente). O processo de mascaramento leva à mudanças na pertinência de alguns pares de *pixels* entre os conjuntos primários. O processo é mostrado na Figura A.3. Dados dois conjuntos quaisquer C_{Origem} e $C_{Destino}$, uma seta desenhada de C_{Origem} para o conjunto $C_{Destino}$ representa a transição de um par de *pixels* pertencente a C_{Origem} que passou para $C_{Destino}$ sob o padrão de modificação expresso pelo valor colocado sobre a aresta em questão. Para cada padrão de modificação $\pi \in \{00, 10, 01, 11\}$ e qualquer subconjunto $A \subset \mathcal{P}$, denote $\rho(\pi, A)$ a probabilidade de um par de *pixels* de A ser alterado pelo padrão π.

Para cada padrão $\pi \in \{00, 10, 01, 11\}$ e cada conjunto primário $A \subset \{W, V, W, Z\}$,

$$\rho(\pi, A) = \rho(\pi, \mathcal{P}). \tag{A.21}$$

A Equação A.21 implica que os *bits* da mensagem estão randomicamente distribuídos no plano LSB da imagem independentemente de qualquer característica da imagem. Desta forma, as relações a seguir são válidas. Seja p a razão de mascaramento dos *pixels* modificados devido ao mascaramento LSB pelo número total de *pixels*:

1. $\rho(00, \mathcal{P}) = (1 - p/2)^2$;
2. $\rho(01, \mathcal{P}) = \rho(10, \mathcal{P}) = p/2(1 - p/2)^2$;
3. $\rho(11, \mathcal{P}) = (p/2)^2$.

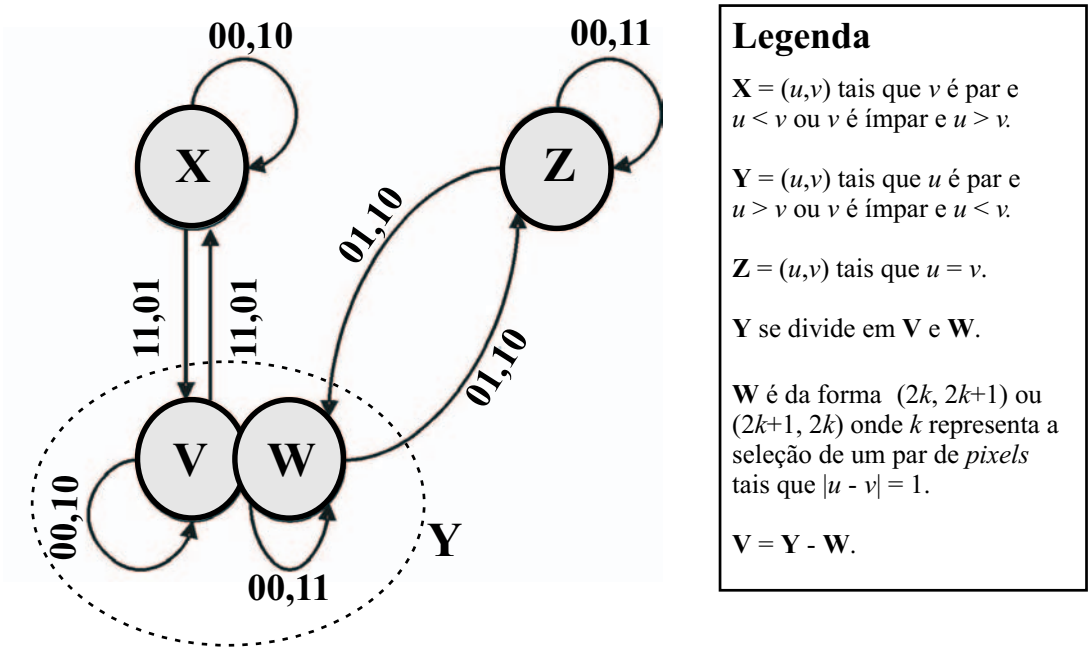


Figura A.3: Diagrama de estados para as transições entre os conjuntos X, V, W, Z devido à inversão LSB.

A partir do que foi exposto e do diagrama de estados da Figura A.3, expressamos as cardinalidades dos conjuntos primários antes e depois do processo de mascaramento como funções de p. Para cada $A \in \{X, Y, V, W, Z\}$, seja A' o conjunto definido da mesma forma que A mas considerando os *pixels* após o processo de mascaramento. Desta forma, obtemos as relações expressas na Equação A.22.

$$\begin{aligned} |X'| &= |X|(1-p/2) + |V|p/2 \\ |V'| &= |V|(1-p/2) + |X|p/2 \\ |W'| &= |W|(1-p+p^2/2) + |Z|p(1-p/2) \end{aligned} \tag{A.22}$$

A partir das relações expressas,

$$|X'| - |V'| = (|X| - |V|)(1-p). \tag{A.23}$$

Dado que, estatisticamente, $|X| = |Y|$ temos que $|X| = |V| + |W|$ e assim

$$|X'| - |V'| = |W|(1-p). \tag{A.24}$$

Observe na Figura A.22 que o processo de mascaramento não altera o conjunto $W \cup Z$. Seja $\gamma = |W| + |Z| = |W'| + |Z'|$. Substituindo $|Z|$ por $\gamma - |W|$,

$$|W'| = |W|(1-p)^2 + \gamma p(1-p/2), \tag{A.25}$$

eliminando $|W|$,

$$|W'| = (|X'| - |V'|)(1-p)^2 + \gamma p(1 - p/2), \tag{A.26}$$

dado que $|X'| + |V'| + |W'| + |Z'| = |\mathcal{P}|$, temos que

$$0.5\gamma p^2 + (2|X'| - |\mathcal{P}|)p + |Y'| - |X'| = 0. \tag{A.27}$$

A Equação A.27 resulta p, a estimativa do tamanho da mensagem escondida na imagem.

Bibliografia

[1] Ismail Avcibas. Steganalysis using image quality metrics. Master's thesis, Department of Computer and Information Science Polytechnic University, Brooklyn, NY, USA, 2002.

[2] Ismail Avcibas, Nasir Memon, and Bülent Sankur. Steganalysis based on image quality metrics. In *Proceedings of the Fourth Workshop on Multimedia Signal Processing*, pages 517–522. IEEE, Oct 2001.

[3] Ismail Avcibas, Nasir Memon, and Bülent Sankur. Steganalysis using image quality metrics. *IEEE Transactions On Image Processing*, 12:221–229, Feb 2003.

[4] Richard E. Bellman. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, USA, 1961. ISBN 0-69107-901-3.

[5] R. W. Buccigrossi and E. P. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions On Image Processing*, 8:1688–1701, 1998.

[6] The Compuserve Group. *Specification of GIF image format*, Jul 1990. `http://www.dcs.ed.ac.uk/home/mxr/gfx/2d/GIF89a.txt`.

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, Sep 1995.

[8] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, first edition, 2000. ISBN 0-52178-019-5.

[9] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, New York, second edition, 2001. ISBN 0-47105-669-3.

[10] Sorina Dumitrescu and Xiaolin Wu. Steganalysis of LSB embedding in multimedia signals. In *Proceedings of the Intl. Conference on Multimedia and Expo*, volume 3, pages 581–584. IEEE, Aug 2002.

[11] Sorina Dumitrescu, Xiaolin Wu, and Nasir Memon. On steganalysis of random LSB embedding in continuous-tone images. In *Proceedings of the Intl. Conference on Image Processing*, volume 3, pages 641–644. IEEE, Jun 2002.

[12] Hany Farid. Detecting steganographic messages in digital images. Technical Report TR2001-412, Department of Computer Science, Dartmouth College, Hanover, NH, USA, Mar 2001.

[13] Hany Farid. Detecting hidden messages using higher-order statistical models. In *Proceedings of the Intl. Conference on Image Processing*, volume 2, pages 905–908. IEEE, Jun 2002.

[14] R. Fisher. The use of multiple measures in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[15] David Freedman, Robert Pisani, and Roger Purves. *Statistics*. George J. McLeod Limited, Toronto, Canadá, first edition, 1978. ISBN 0-39309-076-0.

[16] Jessica Fridrich, Miroslav Goljan, and Rui Du. Detecting LSB steganography in color and grayscale images. *IEEE Multimedia*, 8:22–28, Jan 2001.

[17] Jessica Fridrich, Miroslav Goljan, and Rui Du. Reliable detection of LSB steganography in color and grayscale images. In *Proceedings of ACM Workshop on Multimedia and Security*, pages 27–30, Ottawa, Canada, Oct 2001. ACM.

[18] Jiri Fridrich, Rui Du, and Meng Long. Steganalysis of LSB encoding in color images. In *Proceedings of the Intl. Conference on Multimedia and Expo*, volume 3, pages 1279–1282. IEEE, Aug 2000.

[19] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Springer Verlag, Berlim, Germany, first edition, 2001. ISBN 0-38795-284-5.

[20] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice-Hall, Boston, MA, USA, second edition, 2002. ISBN 0-20118-075-8.

[21] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.

[22] Sara V. Hart, John Ashcroft, and Deborah J. Daniels. Forensic examination of digital evidence: a guide for law enforcement. Technical Report NCJ 199408, U.S. Department of Justice – Office of Justice Programs, Apr 2004.

[23] Neil F. Johnson and Sushil Jajodia. Exploring steganography: Seeing the unseen. *IEEE Computer*, 31:26–34, Feb 1998.

[24] Neil F. Johnson and Sushil Jajodia. Steganalysis of images created using current steganography software. In *Proceedings of the Second Intl. Workshop on Information Hiding*, pages 273–289, London, UK, 1998. Springer-Verlag.

[25] The Joint Expert Photographic Group. *Specification of JPEG image format*, Sep 1992. `http://www.dcs.ed.ac.uk/home/mxr/gfx/2d/JPEG.txt`.

[26] James C. Judge. Steganography: Past, present, future. Technical Report TR552, The SANS Institute, Bethesda, MD, USA, Nov 2001.

[27] David Kahn. *The codebreakers: the comprehensive history of secret communication from ancient times to the internet*. Scribner Inc., New York, NY, USA, revised edition, 1996. ISBN 0-68483-130-9.

[28] Marcus G. Kuhn. The history of steganography. In *Proceedings of the First Intl. Workshop on Information Hiding*, Cambridge, UK, May 1996. Springer–Verlag.

[29] Sun-Yuan Kung, Man-Wai Mak, and Shang-Hung Lin. *Biometric Authentication : A Machine Learning Approach*. Prentice Hall, Boston, MA, USA, 2004. ISBN 0-13147-824-9.

[30] Siwei Lyu. Steganalysis using color wavelet statistics and one-class support vector machines. Master's thesis, Department of Computer Science, Dartmouth College, Hanover, NH, USA, 2002.

[31] Siwei Lyu and Hany Farid. Detecting hidden messages using higher-order statistics and support vector machines. In *Proceedings of the Fifth Intl. Workshop on Information Hiding*, pages 340–354, Noordwijkerhout, The Netherlands, 2002. Springer-Verlag.

[32] Ueli Maurer. A universal statistical test for random bit generators. *Journal of Cryptology*, 5:89–105, Feb 1992.

[33] Rebecca T. Mercuri. The many colors of multimedia security. *Communications of the ACM*, 47:25–29, Dec 2004.

[34] F. C. Mintzer, L. E. Boyle, and A. N. Cases. Toward on-line, worldwide access to vatican library materials. *IBM Journal of Research and Development*, 40:139–162, Mar 1996.

[35] Tom M. Mitchell. *Machine Learning*. WCB/McGraw-Hill, 1997. ISBN 0-07042-807-7.

[36] Sheridan Morris. The future of netcrime now (1) – threats and challenges. Technical Report 62/04, Home Office Crime and Policing Group, 2004.

[37] Bruce Norman. *Secret warfare, the battle of Codes and Ciphers*. Acropolis Books Inc., first edition, 1980. ISBN 0-87491-600-3.

[38] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding — A survey. *Proceedings of the IEEE*, 87:1062–1078, Jul 1999.

[39] Andreas Pfitzmann. Information hiding terminology. In *Proceedings of the First Intl. Workshop on Information Hiding*, Cambridge, UK, May 1996. Springer–Verlag.

[40] Richard Popa. An analysis of steganography techniques. Master's thesis, The "Polytechnic" University of Timisoara, Timisoara, Romênia, 1998.

[41] Niels Provos. Defending against statistical steganalysis. In *Proceedings of the 10^{th} USENIX Security Symposium*, pages 323–336, Washington, DC, USA, Aug 2001. The USENIX Association.

[42] Niels Provos and Peter Honeyman. Detecting steganographic content on the internet. Technical Report CITI 01-11, Department of Computer Science, University of Michigan, Ann Arbor, MI, USA, Nov 2001.

[43] Niels Provos and Peter Honeyman. Hide and seek: an introduction to steganography. *IEEE Security & Privacy Magazine*, 1:32–44, May 2003.

[44] Anderson Rocha. Camaleão: um software para segurança digital utilizando esteganografia. Monografia de final de curso, Depto. de Ciência da Computação, Universidade Federal de Lavras, Dec 2003. Lavras, MG, Brasil.

[45] Gordon Rugg. The mystery of the voynich manuscript. *Scientific American*, June 2004.

[46] Bernhard Schlkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, Boston, MA, USA, first edition, 2001. ISBN 0-26219-475-9.

[47] Li Shi, Sui Ai Fen, and Yang Yi Xian. A LSB steganography detection algorithm. In *Proceedings of the 14^{th} Personal, Indoor and Mobile Radio Communications*, volume 3, pages 2780–2783. IEEE, Sep 2003.

[48] Simon Singh. *O livro dos códigos*. Record, Rio de Janeiro, RJ, Brasil, first edition, 2001. ISBN 8-50105-598-0.

[49] Roman Tzschoppe, Robert Bäuml, Johannes B. Huber, and André Kaup. Steganographic system based on higher-order statistics. In *Proceedings of Fifth Security and Watermarking of Multimedia Contents*, volume 5020, pages 156–166. SPIE, Jun 2003.

[50] P. P. Vaidyanathan. Quadrature mirror filter banks, m-band extensions and perfect reconstruction techniques. *IEEE Signal Processing Magazine*, 4:4–20, Jul 1987.

[51] W. N. Venables and D. M. Smith. *An introduction to R: a programming environment for data analysis and graphics*. R Development Core Team, 2005. ISBN 0-95416-174-2.

[52] W3C. *Specification of PNG image format*, Nov 2003. `http://www.w3.org/TR/PNG/`.

[53] Peter Wayner. *Disappearing cryptography*. Morgan Kaufmann Publishers, San Francisco, CA, USA, second edition, 2002. ISBN 1-55860-769-2.

[54] Andreas Westfeld and Andreas Pfitzmann. Attacks on steganographic systems. In *Proceedings of the Third Intl. Workshop on Information Hiding*, pages 61–76, London, UK, 1999. Springer Verlag.

[55] Philip R. Zimmermann. *The Official PGP User's Guide*. MIT Press, Boston, MA, USA, 1995. ISBN 0-26274-017-6.