Classificadores e Aprendizado em Processamento de Imagens e Visão Computacional

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

Campinas, 3 de março de 2009.

Prof. Dr. Siome Klein Goldenstein (Orientador)

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

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Bibliotecária: Crisllene Queiroz Custódio – CRB8 / 7966

Rocha, Anderson de Rezende

R582c Classificadores e aprendizado em processamento de imagens e visão computacional / Anderson de Rezende Rocha -- Campinas, [S.P. : s.n.], 2009.

Orientador : Siome Klein Goldenstein

Tese (Doutorado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado de máquina - Técnicas. 2. Análise forense de imagens. 3. Esteganálise. 4. Fusão de características. 5. Fusão de classificadores. 6. Classificação multi-classe. 7. Categorização de imagens. I. Goldenstein, Siome Klein. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título

(cqc/imecc)

Título em inglês: Classifiers and machine learning techniques for image processing and computer vision

Palavras-chave em inglês (Keywords): 1. Machine learning - Techniques. 2. Digital forensics. 3. Steganalysis. 4. Feature fusion. 5. Classifier fusion. 6. Multi-class classification. 7. Image categorization.

Área de concentração: Visão Computacional

Titulação: Doutor em Ciência da Computação

Banca examinadora: Prof. Dr. Siome Klein Goldenstein (IC-UNICAMP)
Prof. Dr. Fábio Gagliardi Cozman (Mecatrônica-Poli/USP)
Prof. Dr. Luciano da Fontoura Costa (IF-USP/São Carlos)
Prof. Dr. Ricardo Dahab (IC-UNICAMP)
Prof. Dr. Roberto de Alencar Lotufo (FEEC-UNICAMP)

Data da defesa: 03/03/2009

Programa de Pós-Graduação: Doutorado em Ciência da Computação

TERMO DE APROVAÇÃO

Tese Defendida e Aprovada em 03 de março de 2009, pela Banca examinadora composta pelos Professores Doutores:

Prof. Dr. Fabio Gagliardi Cozman
Departamento de Mecatrônica e Sistemas Mecânicos - Poli / USP.

Prof. Dr. Luciano da Fontoura Costa
Instituto de Física de São Carlos / USP.

Prof. Dr. Ricardo Dahab
IC / UNICAMP.

Prof. Dr. Roberto de Alencar Lotufo
FEEC / UNICAMP.

Prof. Dr. Siome Klein Goldenstein
IC / UNICAMP.

Instituto de Computação
Universidade Estadual de Campinas

Classificadores e Aprendizado em Processamento de Imagens e Visão Computacional

Anderson de Rezende Rocha[1]

3 de março de 2009

Banca Examinadora:

- Prof. Dr. Siome Klein Goldenstein (Orientador)
- Prof. Dr. Fabio Gagliardi Cozman
  Depto. de Mecatrônica e Sistemas Mecânicos (Poli-USP)
- Prof. Dr. Luciano da Fontoura Costa
  Instituto de Física de São Carlos (USP)
- Prof. Dr. Ricardo Dahab
  Instituto de Computação (Unicamp)
- Prof. Dr. Roberto de Alencar Lotufo
  Faculdade de Engenharia Elétrica e de Computação (Unicamp)

[1] Financiado pela *Fundação de Amparo à Pesquisa do Estado de São Paulo* (FAPESP), processo número 05/58103-3.

Resumo

Neste trabalho de doutorado, propomos a utilização de classificadores e técnicas de aprendizado de máquina para extrair informações relevantes de um conjunto de dados (e.g., imagens) para solução de alguns problemas em Processamento de Imagens e Visão Computacional.

Os problemas de nosso interesse são: categorização de imagens em duas ou mais classes, detecção de mensagens escondidas, distinção entre imagens digitalmente adulteradas e imagens naturais, autenticação, multi-classificação, entre outros.

Inicialmente, apresentamos uma revisão comparativa e crítica do estado da arte em análise forense de imagens e detecção de mensagens escondidas em imagens. Nosso objetivo é mostrar as potencialidades das técnicas existentes e, mais importante, apontar suas limitações. Com esse estudo, mostramos que boa parte dos problemas nessa área apontam para dois pontos em comum: a seleção de características e as técnicas de aprendizado a serem utilizadas. Nesse estudo, também discutimos questões legais associadas à análise forense de imagens como, por exemplo, o uso de fotografias digitais por criminosos.

Em seguida, introduzimos uma técnica para análise forense de imagens testada no contexto de detecção de mensagens escondidas e de classificação geral de imagens em categorias como *indoors*, *outdoors*, *geradas em computador* e *obras de arte*.

Ao estudarmos esse problema de multi-classificação, surgem algumas questões: como resolver um problema multi-classe de modo a poder combinar, por exemplo, características de classificação de imagens baseadas em cor, textura, forma e silhueta, sem nos preocuparmos demasiadamente em como normalizar o vetor-comum de características gerado? Como utilizar diversos classificadores diferentes, cada um, especializado e melhor configurado para um conjunto de características ou classes em confusão? Nesse sentido, apresentamos, uma técnica para fusão de classificadores e características no cenário multi-classe através da combinação de classificadores binários. Nós validamos nossa abordagem numa aplicação real para classificação automática de frutas e legumes.

Finalmente, nos deparamos com mais um problema interessante: como tornar a utilização de poderosos classificadores binários no contexto multi-classe mais eficiente e eficaz? Assim, introduzimos uma técnica para combinação de classificadores binários (chamados classificadores base) para a resolução de problemas no contexto geral de multi-classificação.

Abstract

In this work, we propose the use of classifiers and machine learning techniques to extract useful information from data sets (e.g., images) to solve important problems in Image Processing and Computer Vision.

We are particularly interested in: two and multi-class image categorization, hidden messages detection, discrimination among natural and forged images, authentication, and multi-classification.

To start with, we present a comparative survey of the state-of-the-art in digital image forensics as well as hidden messages detection. Our objective is to show the importance of the existing solutions and discuss their limitations. In this study, we show that most of these techniques strive to solve two common problems in Machine Learning: the feature selection and the classification techniques to be used. Furthermore, we discuss the legal and ethical aspects of image forensics analysis, such as, the use of digital images by criminals.

We introduce a technique for image forensics analysis in the context of hidden messages detection and image classification in categories such as *indoors*, *outdoors*, *computer generated*, and *art works*.

From this multi-class classification, we found some important questions: how to solve a multi-class problem in order to combine, for instance, several different features such as color, texture, shape, and silhouette without worrying about the pre-processing and normalization of the combined feature vector? How to take advantage of different classifiers, each one custom tailored to a specific set of classes in confusion? To cope with most of these problems, we present a feature and classifier fusion technique based on combinations of binary classifiers. We validate our solution with a real application for automatic produce classification.

Finally, we address another interesting problem: how to combine powerful binary classifiers in the multi-class scenario more effectively? How to boost their efficiency? In this context, we present a solution that boosts the efficiency and effectiveness of multi-class from binary techniques.

Agradecimentos

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

Estou realizando o sonho de ser doutor. Quando saí de minha pequena cidade aos 10 anos de idade para estudar esse nem parecia ser um sonho alcançável. No entanto, os dia se passaram, e aqui estou hoje amadurecido pelo tempo e agradecendo a ajuda das muitas pessoas com quem cruzei durante a vida. Gostaria de citar 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 esposa Aninha. Vocês são minha inspiração.

Gostaria de estender os agradecimentos ao meu pai Antônio Carlos que, mesmo em sua simplicidade, entende as dificuldades de uma caminhada como essa. Às minhas irmãs Aline e Ana Carla por sempre acreditarem em mim. Agradeço também à Regina Célia pelo apoio constante.

Quero agradecer ao meu orientador Siome Goldenstein. Suas dicas foram muito importantes para o meu crescimento não só como estudante mas também como pesquisador e cidadão. Com você, Siome, aprendi muita coisa, principalmente a fazer as perguntas corretas e a ser crítico.

Estendo meus agradecimentos aos professores da Unicamp com quem tive a oportunidade de ser aluno como: Alexandre Falcão, Anamaria Gomide, Jacques Wainer, Neucimar Leite, Ricardo Anido, Ricardo Panain, Ricardo Torres e Yuzo Iano.

Agradeço aos meus muitos colegas de laboratório por discussões importantes. Em especial agradeço aos diversos colegas com quem tive a oportunidade de ser um colaborador de pesquisa. Agradeço também aos meus colegas de apartamento Luís Meira e Wilson Pavon. Obrigado a todos pela amizade.

Não posso esquecer de mencionar o agradecimento 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 sem o qual eu não poderia me dedicar integralmente à essa pesquisa.

Epígrafe

Every day I remind myself that my inner and outer life are based on the labors of other men, living and dead, and that I must exert myself in order to give in the same measure as I have received and am still receiving.

(*Albert Einstein*)

Dedicatória

Dedico este trabalho à minha mãe **Lucília**. Mãe, os pobres também podem. Dedico também à minha esposa **Aninha**, por ser minha “principeza”. Aninha, você é um presente para mim. Isso é para vocês.

Sumário

Capítulo 1

Introdução

Em Processamento de Imagens e Visão Computacional, muitas vezes, a solução de determinados problemas pode exigir o correto entendimento do contexto da cena analisada ou mesmo das inter-relações compartilhadas por cenas de um mesmo grupo semântico. No entanto, definir precisamente as nuanças e características que gostaríamos de selecionar não é uma tarefa fácil. Nesse contexto, técnicas de aprendizado de máquina e reconhecimento de padrões podem tornar-se ferramentas valiosas.

A extração de características representativas de um conjunto de dados (e.g., imagens) é uma tarefa complexa, e exige modelos sofisticados. Não há uma forma única e sistemática para extrair características ou relações métricas entre exemplos. Como passo inicial, podemos utilizar duas abordagens principais: *generativa* e *discriminativa* [146].

Com a *abordagem generativa*, procuramos resolver um problema dando ênfase no processo de geração dos dados sob análise. Normalmente, modelamos o sistema como uma distribuição conjunta de probabilidade (*Joint probability function*) e, desta forma, podemos criar exemplos artificiais que podem ser inseridos no sistema. Exemplos de modelos que utilizam a abordagem generativa são: Classificadores Bayesianos, *Markov Random Fields* e *Gaussian Mixture Models* [55]. Por outro lado, na *abordagem discriminativa*, procuramos encontrar as fronteiras que melhor separam um conjunto de classes do nosso problema. Classificadores como *Support Vector Machines* (SVMs) [31] utilizam esta abordagem. Para entender melhor, considere a Figura 1.1. Nesse problema de classificação, a abordagem generativa objetiva encontrar relações métricas na classe dos círculos ($+1$) e dos triângulos (-1), de modo a modelar o processo de geração desses dados. Em contrapartida, a abordagem discriminativa procura modelar a melhor fronteira de separação das duas classes.

De forma geral, os modelos generativo e discriminativo podem variar de acordo com cada aplicação. Nesta pesquisa de doutorado, nós avaliamos e aplicamos a melhor abordagem de acordo com o problema analisado. Em alguns casos, pode ser necessário utilizar uma associação destas duas abordagens [69] construindo um modelo de extração/classificação de características mais robusto.

A associação de informações aprendidas a partir de um conjunto de dados não é uma idéia

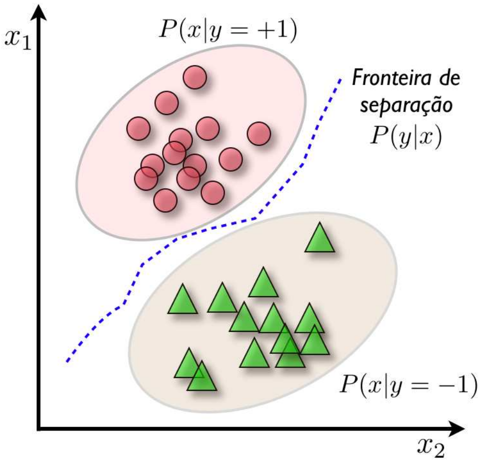


Figura 1.1: Diferentes abordagens de solução de problemas em Aprendizado de Máquina: generativa e discriminativa.

nova. Muito se tem pesquisado para descobrir como nós humanos interpretamos uma determinada cena e como podemos extrair informações de nossa interpretação de modo que possamos associá-las na resolução de certos problemas.

Viola e Jones [182] apresentaram uma abordagem descritiva para detecção de faces através da codificação de características que demonstram um domínio de conhecimentos *ad hoc* das imagens analisadas. Os autores extraem informações das imagens a partir de classificadores bem simples dispostos em um modelo de cascata. Esta abordagem mostrou-se mais eficiente que sistemas baseados em informações locais (*pixels*).

Lyu e Farid [99] apresentaram uma técnica que decompõe uma imagem em um modelo de posição espacial, orientação e escala capaz de fornecer descritores que podem ser utilizados para extrair modelos artísticos de um determinado conjunto de obras de um certo artista. A partir do aprendizado dessas informações, pode-se traçar o perfil do artista sendo analisado.

Nesta tese de doutorado, organizada na forma de coletânea de artigos, propomos a utilização de classificadores e técnicas de aprendizado de máquina para extrair informações relevantes de um conjunto genérico de dados (e.g., imagens), similaridade entre um certo conjunto de imagens ou dados, ou mesmo sua percepção semântica, para solução de alguns problemas em Processamento de Imagens e Visão Computacional.

Os problemas de nosso interesse são: categorização de imagens em duas ou mais classes, detecção de mensagens escondidas, distinção entre imagens digitalmente adulteradas e imagens naturais, autenticação, multi-classificação, entre outros.

Inicialmente, nos Capítulos 2 e 3, apresentamos uma revisão comparativa e crítica do estado da arte em análise forense de imagens e detecção de mensagens escondidas em imagens. Nosso objetivo é mostrar as potencialidades das técnicas existentes e, mais importante, apontar suas limitações. Com esse estudo, mostramos que boa parte dos problemas nessa área apontam para dois pontos em comum: a seleção de características e as técnicas de aprendizado a serem utilizadas. Nesse estudo, também discutimos questões legais associadas à análise forense de imagens como, por exemplo, o uso de fotografias digitais por criminosos.

Em seguida, no Capítulo 4, introduzimos uma técnica para análise forense de imagens testada no contexto de detecção de mensagens escondidas e de classificação geral de imagens em categorias como *indoors*, *outdoors*, *geradas em computador* e *obras de arte*.

Ao estudarmos esse problema de multi-classificação, surgem algumas questões: como resolver um problema multi-classe de modo a poder combinar, por exemplo, características de classificação de imagens baseadas em cor, textura, forma e silhueta, sem nos preocuparmos demasiadamente em como normalizar o vetor-comum de características gerado? Como utilizar diversos classificadores diferentes, cada um, especializado e melhor configurado para um conjunto de características ou classes em confusão? Nesse sentido, no Capítulo 5, apresentamos uma técnica para fusão de classificadores e características no cenário multi-classe através da combinação de classificadores binários. Nós validamos nossa abordagem numa aplicação real para classificação automática de frutas e legumes.

Finalmente, nos deparamos com mais um problema interessante: como tornar a utilização de poderosos classificadores binários no contexto multi-classe mais eficiente e eficaz? Assim, no Capítulo 6, introduzimos uma técnica para combinação de classificadores binários (chamados classificadores base) para a resolução de problemas no contexto geral de multi-classificação.

No restante do Capítulo 1, apresentamos um resumo de nossas contribuições nesse trabalho de doutorado. Os capítulos posteriores apresentam mais detalhes sobre cada uma das contribuições. Antes de cada capítulo, apresentamos um breve resumo, em português, sobre o assunto a ser tratado e, em seguida, apresentamos o capítulo em inglês. Ao final, apresentamos as considerações finais de nosso trabalho.

1.1 Detecção de adulterações em imagens digitais

Ao campo de pesquisas relacionado à análise de imagens para verificação de sua autenticidade e integridade denominamos *Análise Forense de Imagens*. Com o advento da *internet* e das câmeras de alta performance e de baixo custo juntamente com poderosos pacotes de *software* de edição de imagens (Photoshop, Adobe Illustrator, Gimp), usuários comuns tornaram-se potenciais especialistas na criação e manipulação de imagens digitais. Quando estas modificações deixam de ser inocentes e passam a implicar em questões legais, torna-se necessário o desenvolvimento de abordagens eficientes e eficazes para sua detecção.

A identificação de imagens que foram digitalmente adulteradas é de fundamental importância atualmente [43,138,141]. O julgamento de um crime, por exemplo, pode estar sendo baseado em evidências que foram fabricadas especificamente para enganar e mudar a opinião de um júri. Um político pode ter a opinião pública lançada contra ele por ter aparecido ao lado de um traficante procurado mesmo sem nunca ter visto este traficante antes.

No Capítulo 2, apresentamos um estudo crítico das principais técnicas existentes na análise forense de imagens. No Capítulo 3, mostramos mais especificamente algumas técnicas para o mascaramento digital de informações e para a detecção de mensagens escondidas em imagens. Nos dois capítulos, mostramos que boa parte dos problemas relacionados à análise forense de imagens apontam para dois pontos em comum: a seleção de características e as técnicas de aprendizado a serem utilizadas.

Como discutimos nos Capítulos 2 e 3, atualmente, não existem metodologias estabelecidas para verificar a autenticidade e integridade de imagens digitais de forma automática. Embora a marcação digital (*watermarking*) possa ser utilizada em algumas situações, sabemos que a grande maioria das imagens digitais não possui marcação. Adicionalmente, qualquer solução baseada em marcação digital implicaria na implementação de tal abordagem diretamente nos sensores de aquisição das imagens o que tornaria seu uso restritivo. Além disso, possivelmente haveria perdas na qualidade do conteúdo da imagem devido à inserção das marcações.

De forma geral, as técnicas propostas na literatura para análise forense de imagens são categorizadas em quatro grandes áreas de acordo com o seu foco principal (c.f., Cap. 2 e 3): (1) identificação da origem da imagem; (2) distinção entre imagens naturais e imagens sintéticas; (3) detecção de mensagens escondidas; e (4) detecção de falsificação em imagens.

1. **Identificação da origem da imagem.** Consiste no conjunto de técnicas para investigar e identificar as características do dispositivo de captura de uma imagem (e.g., câmera digital, *scanner*, gravadora). Para estas técnicas, normalmente esperamos dois resultados: (1) a classe ou modelo da fonte utilizada e (2) as características da fonte específica utilizada.

2. **Identificação de imagens sintéticas.** Consiste no conjunto de técnicas para investigar e identificar as características que possam classificar uma imagem como falsa (não natural).

3. **Detecção de mensagens escondidas.** Consiste no conjunto de técnicas para a detecção de mensagens escondidas em imagens digitais. Tipicamente, essas mensagens são inseridas através da modificação de propriedades das imagens (e.g., *pixels*).

4. **Identificação de adulterações.** Consiste na detecção de adulterações em imagens digitais. Tipicamente, uma imagem (ou parte dela) sofre uma ou mais manipulações digitais tais como: operações afins (e.g., aumento, redução, rotação), compensação de cor e brilho, supressão de detalhes (e.g., filtragem, adição de ruído, compressão).

Resultados obtidos

A análise crítica que apresentamos no Capítulo 2 é uma compilação de nosso trabalho submetido ao *ACM Computing Surveys*. O banco de dados de imagens que discutimos nesse capítulo é resultado de nosso artigo [154] no *IEEE Workshop on Vision of the Unseen* (WVU). Ambos os trabalhos foram produzidos com a colaboração dos pesquisadores Walter J. Scheirer e Terrance E. Boult da Universidade do Colorado em Colorado Springs. Finalmente, o trabalho apresentado no Capítulo 3 é o resultado de nosso artigo [149] na *Revista de Informática Teórica e Aplicada* (RITA).

1.2 Esteganálise e categorização de imagens

Pequenas perturbações feitas nos canais menos significativos de imagens digitais (e.g., canal LSB) são imperceptíveis aos humanos mas são estatisticamente detectáveis no contexto de análise de imagens [147, 188].

Nesse sentido, no Capítulo 4, apresentamos uma abordagem para meta-descrição de imagens denominada Randomização Progressiva (PR[1]) para nos auxiliar nos problemas de: (1) Detecção de mensagens escondidas em imagens digitais; e (2) Categorização de imagens.

1.2.1 Detecção de mensagens escondidas

Neste problema, procuramos aperfeiçoar e dar robustez ao trabalho desenvolvido em meu mestrado [35]. Estudamos e desenvolvemos técnicas capazes de permitir a detecção de mensagens escondidas em imagens digitais.

Grande parte das técnicas de *esteganografia*, a arte das comunicações escondidas, possuem falhas e/ou inserem artefatos (padrões) detectáveis nos objetos de cobertura (utilizados para esconder uma determinada mensagem). A identificação destes artefatos e sua correta utilização na detecção de mensagens escondidas constituem a arte e a ciência conhecida como *esteganálise* [149].

O método de randomização progressiva proposto permite a detecção de mensagens escondidas em imagens com compressão sem perdas (e.g., PNGs). Além disso, o método permite apontar quais os tipos de imagens são mais sensíveis ao mascaramento de mensagens bem como quais tipos de imagens são mais propícios a este tipo de operações.

1.2.2 Categorização de imagens – Cenário de duas classes

O conhecimento semântico sobre uma determinada mídia nos permite desenvolver técnicas inteligentes de processamento dessas mídias baseadas em seu conteúdo. Câmeras digitais ou aplicações de computador podem corrigir cor e brilho automaticamente levando em consideração propriedades da cena analisada. Nesses casos, informações locais das mídias podem ser insuficientes para determinados problemas.

Nesse trabalho de doutorado, procuramos desenvolver uma técnica capaz de associar informações coletadas através de relações encontradas em um grande banco de dados de imagens para separar imagens naturais de imagens geradas em computador [37, 98, 119], imagens em ambiente externo (*outdoors*) de imagens em ambiente interno (*indoors*) [92, 128, 162], e imagens naturais de imagens de obras de arte [34]. Nossa abordagem consiste em capturar propriedades estatísticas das duas classes analisadas de cada vez e buscar diferenças nestas propriedades.

1.2.3 Categorização de imagens – Cenário multi-classe

Denomina-se categorização de imagens ao conjunto de técnicas que distinguem classes de imagens, apontando o tipo de uma imagem. Nesse problema, objetivamos desenvolver uma abor-

[1] Originalmente, denominamos nosso meta-descritor como *Progressive Randomization (PR).*

dagem de categorização de imagens para as classes *indoors*, *outdoors*, *geradas em computador* e *artes*. Não consideramos classes específicas de objetos tais como carros ou pessoas. Um cenário típico para uma aplicação é o agrupamento de fotos em álbuns automaticamente de acordo com classes. A solução que apresentamos é simples, unificada e relativamente possui baixa dimensionalidade[2].

1.2.4 Randomização Progressiva (PR)

PR é um novo meta-descritor que captura as diferenças entre classes gerais de imagens usando os artefatos estatísticos inseridos durante um processo de perturbação sucessiva das imagens analisadas. Nossos experimentos demonstraram que esta técnica captura bem a separabilidade de algumas classes de imagens. A observação mais importante é que classes diferentes de imagens possuem comportamentos distintos quando submetidas a sucessivas perturbações. Por exemplo, um conjunto de imagens que não possui mensagens escondidas apresenta diferentes artefatos mediante sucessivas perturbações que um conjunto de imagens que possui mensagens escondidas.

No Algoritmo 1, resumimos os quatro passos principais da Randomização Progressiva aplicada à Esteganálise e à Categorização de Imagens. Os quatro passos são: (1) o processo de randomização; (2) seleção de regiões características; (3) descrição estatística; e (4) invariância.

Algorithm 1 Meta-descritor de Randomização Progressiva (PR).

```
Require: Imagem de entrada $I$; Porcentagens $P = \{P_1, \dots P_n\}$;
1: Randomização: faça $n$ perturbações nos bits menos significativos de $I$
       $$\{O_i\}_{i=0\dots n.} = \{I, T(I, P_1), \dots, T(I, P_n)\}.$$
2: Seleção de regiões: selecione $r$ regiões de cada imagem $i \in \{O_i\}_{i=0\dots n}$
       $$\{O_{ij}\}_{\substack{i=0\dots n,\\ j=1\dots r.}} = \{O_{01}, \dots, O_{nr}\}.$$
3: Descrição estatística: calcule $m$ descritores estatísticos para cada região
       $$\{d_{ijk}\} = \{d_k(O_{ij})\}_{\substack{i=0\dots n,\\ j=1\dots r,\\ k=1\dots m.}}$$
4: Invariância: normalize os descritores de acordo com seus valores na imagem de entrada $I$
       $$\mathbf{F} = \{f_e\}_{e=1\dots n\times r\times m} = \left\{\frac{d_{ijk}}{d_{0jk}}\right\}_{\substack{i=0\dots n,\\ j=1\dots r,\\ k=1\dots m.}},$$
5: Use as características $\{d_{ijk}\} \in \mathbb{R}^{(n+1)\times r\times m}$ (não-normalizadas) ou $\{d_{ijk}\} \in \mathbb{R}^{n\times r\times m}$ (normalizadas) em seu classificador de padrões favorito.
```

[2] Baixa dimensionalidade refere-se a um baixo número de características no processo de descrição dos elementos analisados.

Perturbação dos pixels

Seja $\mathbf{x}$ uma variável aleatória com distribuição de Bernoulli com probabilidade $Prob\{\mathbf{x} = 0\}) = Prob(\{\mathbf{x} = 1\}) = \frac{1}{2}$, B uma seqüência de *bits* composta por ensaios independentes de $\mathbf{x}$, p uma porcentagem, e S um conjunto aleatório de *pixels* em uma imagem de entrada.

Dada uma imagem de entrada I com $|I|$ *pixels*, nós definimos uma perturbação $T(I, p)$ no canal de *bits* menos significativo (LSB) como o processo de substituição dos LSBs de S de tamanho $p \times |I|$ de acordo com a seqüência de *bits* B.

Considere um *pixel* $px_i \in S$ e um *bit* associado $b_i \in B$

$$\mathcal{L}(px_i) \leftarrow b_i \text{ para todo } px_i \in S. \tag{1.1}$$

onde $\mathcal{L}(px_i)$ é o LSB do *pixel* px_i. A Figura 1.2 mostra um exemplo de uma perturbação usando os *bits* $B = 1110$.

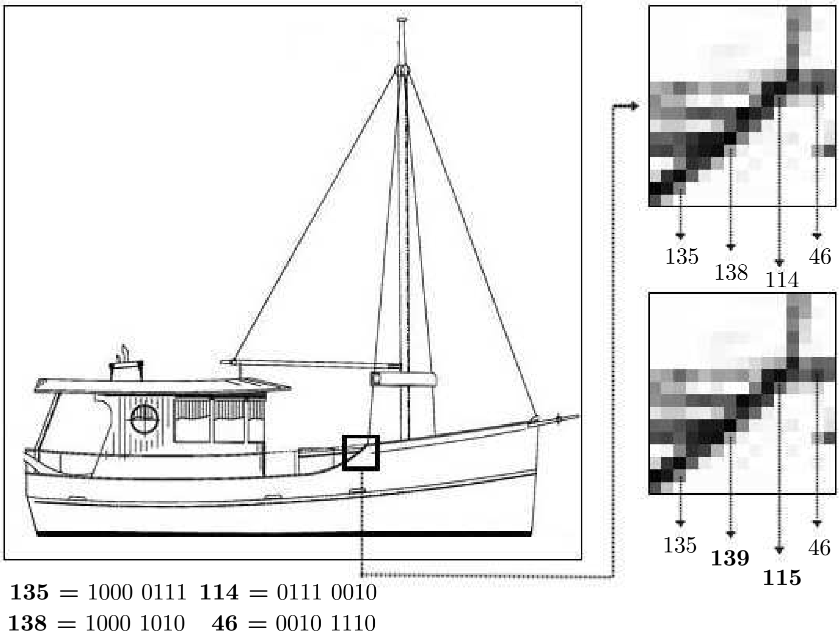


Figura 1.2: Um exemplo de perturbação LSB usando os *bits* $B = 1110$.

O processo de randomização

Dado uma imagem original I como entrada, o processo de randomização consiste na aplicação sucessiva de perturbações $T(I, P_1), \ldots, T(I, P_n)$ nos LSBs dos *pixels* de I. O processo retorna n imagens que apenas diferem entre si nos canais LSBs usados nas perturbações e são idênticas ao olho nu.

As $T(I, P_i)$ transformações são perturbações de diferentes porcentagens (pesos) nos LSBs disponíveis. Em nosso trabalho base, utilizamos $n = 6$ onde $P = \{1\%, 5\%, 10\%, 25\%, 50\%, 75\%\}$, $P_i \in P$ denota os tamanhos relativos dos conjuntos de *pixels* selecionados S.

Seleção de regiões

Propriedades locais não aparecem diretamente sob uma investigação global [188]. Nós utilizamos descritores estatísticos em regiões locais para capturar as mudanças inseridas pelas perturbações sucessivas (c.f., Sec. 1.2.4).

Dada uma imagem I, nós usamos r regiões com tamanho $l \times l$ *pixels* para produzir descritores estatísticos localizados. Na Figura 1.3, nós mostramos uma configuração com $r = 8$ regiões com sobreposição de informações.

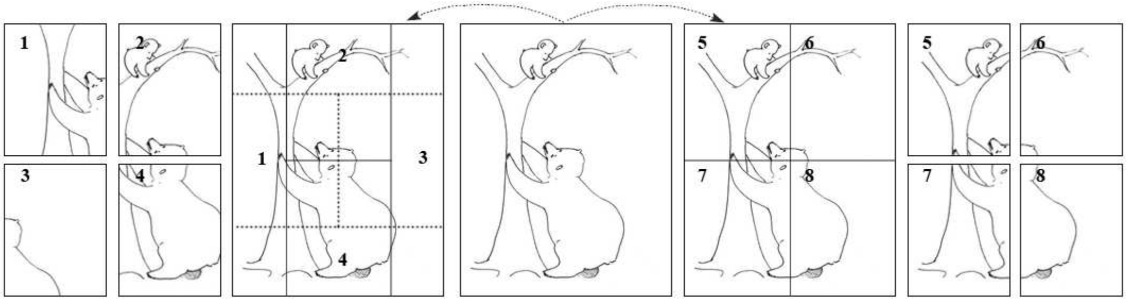


Figura 1.3: Oito regiões de interesse considerando sobreposição de informações.

Descrição estatística

As perturbações LSB mudam o conteúdo de um conjunto selecionado de *pixels* e induzem mudanças localizadas nas estatísticas dos *pixels*. Um *pixel* com L *bits* possui 2^L valores possíveis e representa 2^{L-1} classes de invariância se consideramos possíveis mudanças apenas no canal LSB (c.f., Sec. 1.2.4). Chamamos estas classes de invariância de pares de valores (PoV[3]).

Quando perturbamos todos os LSBs disponíveis em S com uma seqüência B, a distribuição de valores 0/1 de um PoV será a mesma de B. A análise estatística compara os valores teóricos esperados com os observados dos PoVs após o processo de perturbação.

Nós aplicamos os descritores estatísticos χ^2 (Teste do Chi-quadrado) [191] e U_T (Teste Universal de Ueli Maurer) [102] para analisar estas mudanças.

[3] Pair of Values.

Invariância

Em algumas situações, é necessário usar um descritor de características invariante. Para tal, usamos a taxa de variação de nossos descritores estatísticos em relação a cada perturbação sucessiva, ao invés de seus valores diretos. Nós normalizamos todos os valores de descritores decorrentes das transformações em relação aos seus valores na imagem de entrada (sem perturbação)

$$F = \{f_e\}_{e=1\ldots n\times r\times m} \quad = \quad \left\{\frac{d_{ijk}}{d_{0jk}}\right\}_{\substack{i=0\ldots n,\\ j=1\ldots r,\\ k=1\ldots m.}} \quad , \tag{1.2}$$

onde d denota um descritor $1 \leq k \leq m$ de uma região $1 \leq j \leq r$ de uma imagem $0 \leq i \leq n$ e F é o vetor de características final gerado para a imagem I.

A necessidade da etapa de invariância depende da aplicação. Por exemplo, ela é necessária no contexto de detecção de mensagens escondidas uma vez que queremos diferenciar imagens que contêm mensagens escondidas daquelas que não contêm. A classe das imagens não é relevante. No contexto de categorização de imagens, os valores em si são mais importantes que a taxa de variabilidade em perturbações sucessivas.

1.2.5 Resultados obtidos

O Capítulo 4 é uma compilação de nosso trabalho submetido à *Elsevier Computer Vision and Image Understanding* (CVIU). Após um estudo que mostrou viabilidade comercial de nossa técnica, conseguimos o depósito de uma patente nacional[4] junto ao INPI[5] e sua versão internacional[6] junto ao PCT[7].

Finalmente, o trabalho de detecção de mensagens nos rendeu a publicação [147] no *IEEE Intl. Workshop on Multimedia and Signal Processing (MMSP)*. A extensão da técnica para o cenário multi-classe (*indoors*, *outdoors*, *geradas em computador*, e *obras de arte*) resultou o artigo [148] no *IEEE Intl. Conference on Computer Vision (ICCV)*.

1.3 Fusão multi-classe de características e classificadores

Algumas vezes, problemas de categorização multi-classe são complexos e a fusão de informações de vários descritores torna-se importante.

Embora a fusão de características seja bastante eficaz para alguns problemas, ela pode produzir resultados inesperados quando as diferentes características não estão normalizadas e preparadas de forma adequada. Além disso, esse tipo de combinação tem a desvantagem de aumentar o número de características do vetor base de descrição o que, por sua vez, pode levar à necessidade de mais elementos para o treinamento.

[4] http://www.inovacao.unicamp.br/report/patentes_ano2006-inova.pdf

[5] Instituto Nacional de Propriedade Industrial.

[6] http://www.inovacao.unicamp.br/report/inte-allpatentes2007-unicamp071228.pdf

[7] Patent Cooperation Treaty.

Além disso, em certas ocasiões, alguns classificadores produzem melhores resultados para determinados descritores do que para outros. Isto sugere que a combinação de classificadores em problemas multi-classe, cada um especializado em um caso particular, pode ser interessante.

Embora a combinação de classificadores e características não seja tão direta no cenário multi-classe, ela é um problema simples para problemas de classificação binários. Nesse caso, é possível combinar diferentes classificadores e características usando regras simples de fusão tais como `and`, `or`, `max`, `sum`, ou `min` [16]. No entanto, para problemas multi-classe, a fusão torna-se um pouco mais complicada dado que uma característica pode apontar como resultado a classe C_i, outra característica apontar a classe C_j, e ainda outra poderia produzir o resultado C_k. Com muitos resultados diferentes para um mesmo exemplo de teste, torna-se difícil definir uma política consistente para combinar as características selecionadas.

Uma abordagem muito usada consiste na combinação dos vetores característicos em um grande vetor de descrição. Embora bem eficaz em alguns casos, esta abordagem pode, também, produzir resultados inesperados quando o vetor não é normalizado e preparado da forma adequada. Em primeiro lugar, para criar o vetor combinado de características, precisamos lidar com a natureza diferente de cada vetor característico. Alguns podem ser bem condicionados possuindo apenas variáveis contínuas e limitadas, outros podem ser mal-condicionados para essa combinação tais como aqueles que possuem variáveis categóricas. Adicionalmente, algumas variáveis podem ser contínuas e não limitadas. Em resumo, para unificar todas as características, precisamos de um pré-processamento e normalização adequados. Entretanto, algumas vezes esse pré-processamento é trabalhoso.

Esse tipo de combinação de características eventualmente pode levar à maldição da dimensionalidade. Dado que temos mais dimensões no vetor característico combinado, precisamos de mais exemplos de treinamento.

Finalmente, se precisarmos adicionar mais uma característica àquelas existentes, temos que pré-processar os dados novamente para uma nova normalização.

1.3.1 Solução proposta

No Capítulo 5, nós apresentamos uma abordagem para combinar classificadores e características capaz de lidar com a maior parte dos problemas citados anteriormente. Nosso objetivo é combinar um conjunto de características e os classificadores mais apropriados para cada uma de modo a melhorar a performance sem comprometer a eficiência.

Nós propomos lidar com um problema multi-classe a partir da combinação de um conjunto de classificadores binários. Podemos definir a binarização de classes como um mapeamento de um problema multi-classe para vários problemas binários (dividir para conquistar) e a subsequente combinação de seus resultados para derivar a predição multi-classe. Nos referimos aos classificadores binários como classificadores base. A binarização de classes têm sido utilizada na literatura para estender classificadores naturalmente binários tais como SVM para multi-classe [5,38,115]. Entretanto, de acordo com nosso conhecimento, esta abordagem não foi utilizada anteriormente para a fusão de classificadores e características.

Para entender a binarização de classes, considere um problema com três classes. Nesse caso, uma binarização simples consiste no treinamento de três classificadores binários. Nesse sentido, nós precisamos $O(N^2)$ classificadores base, onde N é o número de classes.

Nós treinamos o ij^{esimo} classificador binário utilizando os padrões da classe i como positivos e os padrões da classe j como negativos. Para obter o resultado final, calculamos a distância mínima do vetor binário gerado para o padrão binário que representa cada classe.

Considere novamente o exemplo com três classes como mostramos na Figura 1.4. Nesse exemplo, nós temos as classes: *Triângulos* $\triangle$, *Círculos* $\bigcirc$, e *Quadrados* $\square$. Claramente, uma primeira característica que podemos usar para categorizar os elementos dessas classes pode ser baseado na forma. Podemos também utilizar propriedades de cor e textura. Para resolver esse problema, treinamos alguns classificadores binários diferenciando duas classes por vez, tais como: $\triangle \times \bigcirc$, $\triangle \times \square$, e $\bigcirc \times \square$. Adicionalmente, nós representamos cada uma das classes com um identificador único ($\triangle = \langle +1, +1, 0 \rangle$).

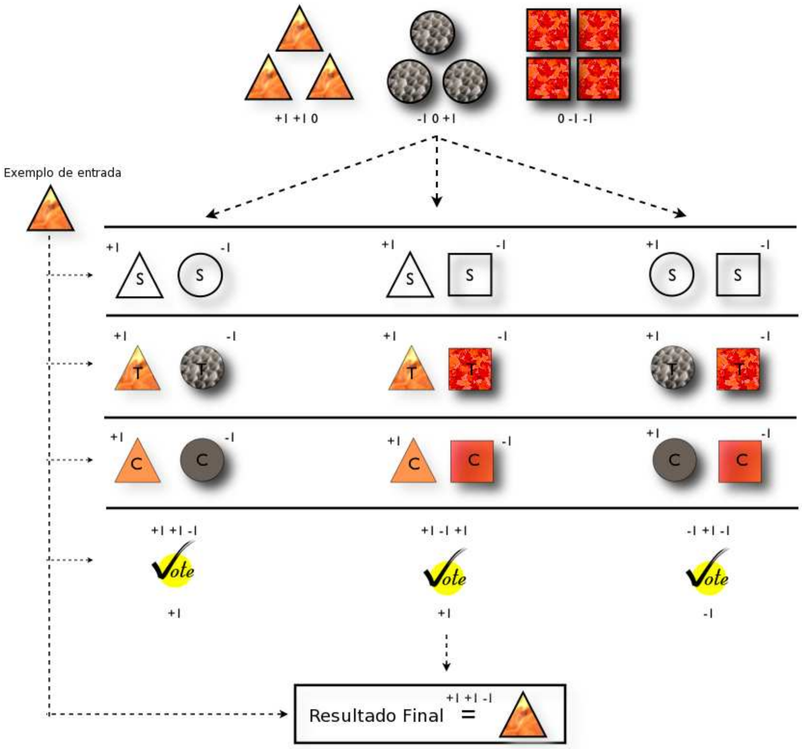


Figura 1.4: Pequeno exemplo para combinação de classificadores e características.

Ao recebermos um exemplo para classificar, digamos um com a forma de triângulo, como mostramos na Figura 1.4, primeiro aplicamos nossos classificadores binários para verificar se o exemplo testado é um triângulo ou um círculo baseado na forma, textura e cor. Cada classificador nos dá uma resposta binária. Por exemplo, digamos que nosso resultado seja os votos $\langle +1, +1, -1 \rangle$ para o classificador binário $\triangle \times \bigcirc$. Dessa forma, nós podemos usar o voto majoritário e selecionar uma resposta (+1, neste caso, ou $\triangle$). Então, repetimos o procedimento e testamos se o exemplo analisado é um triângulo ou um quadrado para cada uma das características de interesse. Finalmente, depois de efetuar o último teste, temos como resultado um vetor binário. Basta então calcularmos o mínima distância deste vetor aos vetores identificadores de cada classe. Nesse exemplo, a resposta final é dada pela mínima distância de

$$\min dist(\langle 1,1,-1\rangle, \{\langle 1,1,0\rangle, \langle -1,0,1\rangle, \langle 0,-1,-1\rangle\}). \qquad (1.3)$$

Um aspecto importante dessa abordagem é que ela requer mais armazenamento dado que após o treinamento dos classificadores binários nós precisamos armazenar seus parâmetros. Dado que nós analisamos mais características, precisamos de mais espaço. Com respeito ao tempo de execução, tem também um crescimento dado que precisamos testar mais classificadores binários para obter uma resposta. Entretanto, muitos classificadores em nosso dia-a-dia empregam algum tipo de binarização de classes (e.g., SVMs). Além disso, como apresentamos no Capítulo 6, existem soluções efetivas para combinar tais classificadores binários de forma eficiente.

Embora precisemos de mais espaço de armazenamento, a abordagem apresentada tem as seguintes vantagens:

1. Com a combinação independente de características, temos mais confiança na resposta produzida dado que ela é calculada a partir de mais de uma simples característica. Dessa forma, temos um mecanismo simples de correção de erros que pode resistir à algumas classificações erradas;

2. Podemos desenvolver classificadores e características específicas para separar classes em confusão;

3. Podemos selecionar as características que realmente são importantes na fusão. Esse procedimento não é direto quando temos apenas um grande vetor de características combinadas.

4. A adição de novas classes requer apenas o treinamento para os novos classificadores binários relacionados àquelas classes.

5. A adição de novas características é simples e requer apenas treinamento parcial.

6. Como não aumentamos o tamanho de nenhum vetor de características, temos menor probabilidade de sofrermos da maldição da dimensionalidade, não necessitando, portanto, adicionar mais exemplos de treinamento quando combinando mais características.

Finalmente, nós validamos nossa abordagem de fusão de classificadores e características numa aplicação real para categorização automática de frutas e legumes, como apresentamos no Capítulo 5.

1.3.2 Resultados obtidos

O Capítulo 5 é uma compilação de nosso trabalho submetido à *Elsevier Computers and Electronics in Agriculture* (Compag) e do artigo [153] no *Brazilian Symposium of Computer Graphics and Image Processing* (Sibgrapi). Esses trabalhos foram produzidos com a colaboração dos pesquisadores Daniel C. Hauagge e Jacques Wainer do Instituto de Computação da Unicamp.

1.4 Multi-classe a partir de classificadores binários

Muitos problemas reais de reconhecimento e de classificação freqüentemente necessitam mapear várias entradas em uma dentre centenas ou milhares de possíveis categorias. Muitos pesquisadores têm proposto técnicas efetivas para classificação de duas classes nos últimos anos. No entanto, alguns classificadores poderosos tais como SVMs são difíceis de estender para o cenário multi-classe. Em tais casos, a abordagem mais comum é a de reduzir a complexidade do problema multi-classe para pequenos e mais simples problemas binários (dividir para conquistar) [38, 82, 127, 145].

Ao utilizar classificadores binários com algum critério final de combinação (redução de complexidade), muitas abordagens descritas na literatura partem do princípio de que os classificadores binários utilizados na classificação são independentes e aplicam um sistema de votação como política final de combinação. Entretanto, a hipótese da independência não é a melhor escolha em todos os casos.

Nesse trabalho, nós abordamos o problema de classificação multi-classe apresentando uma forma efetiva de agrupar dicotomias altamente correlacionadas (não supondo independência entre todas elas). Nós denominamos a técnica de *Affine-Bayes* (c.f., Sec. 1.4.1).

1.4.1 Affine-Bayes

Apresentamos, a seguir, nossa abordagem generativa Bayesiana para multi-classificação. Um problema multi-classe típico resolvido a partir da combinação de classificadores binários possui três etapas básicas [145]: (1) a criação da matriz de codificação dos classificadores; (2) a escolha dos classificadores binários base; e (3) a estratégia de decodificação. A solução que propomos enquadra-se, principalmente, na parte 3.

Considerando a etapa de decodificação, nós introduzimos o conceito de relações afins entre classificadores binários e apresentamos uma abordagem efetiva para achar grupos de classificadores binários altamente correlacionados. Finalmente, apresentamos duas novas estratégias: uma para reduzir o número necessário de dicotomias na classificação multi-classe e a outra para achar novas dicotomias para substituir aquelas menos discriminativas. Esses dois procedimentos podem ser utilizados iterativamente para complementar a abordagem básica de *Affine-Bayes* e melhorar a performance geral de classificação.

Para classificar uma determinada entrada, nós usamos um time de classificadores binários base $\mathcal{T}$. Nós chamamos $\mathcal{O}_{\mathcal{T}}$ uma realização de $\mathcal{T}$. Cada elemento de $\mathcal{T}$ é um classificador binário base (dicotomia) e produz uma saída $\in \{-1, +1\}$.

Dado uma entrada x para classificar, uma realização de $\mathcal{O}_\mathcal{T}$ contem a informação para determinar a classe de x. Em outras palavras, $P(y = c_i|x) = P(y = c_i|\mathcal{O}_\mathcal{T})$.

No entanto, não temos a probabilidade $P(y = c_i|\mathcal{O}_\mathcal{T})$. Pelo teorema de Bayes, temos que

$$\begin{aligned} P(y = c_i|\mathcal{O}_\mathcal{T}) &= \frac{P(\mathcal{O}_\mathcal{T}|y = c_i)P(y = c_i)}{P(\mathcal{O}_\mathcal{T})} \\ &\propto P(\mathcal{O}_\mathcal{T}|y = c_i)P(y = c_i) \end{aligned} \tag{1.4}$$

$P(\mathcal{O}_\mathcal{T})$ é apenas um fator de normalização e pode ser eliminado.

Abordagens anteriores resolveram o modelo acima considerando independência entre todas as dicotomias no time $\mathcal{T}$ [127]. Se considerarmos independência entre todas as dicotomias, o modelo na Equação 1.4 se torna

$$P(y = c_i|\mathcal{O}_\mathcal{T}) \propto \prod_{t \,\in\, \mathcal{T}} P(\mathcal{O}^t_\mathcal{T}|y = c_i)P(y = c_i), \tag{1.5}$$

e a classe da entrada x é dada por

$$cl(x) = \arg\max_i \prod_{t \,\in\, \mathcal{T}} P(\mathcal{O}^t_\mathcal{T}|y = c_i)P(y = c_i). \tag{1.6}$$

Embora a restrição de independência simplifique o modelo, ela impõe várias limitações e não é a melhor escolha em todos os casos. Em geral, é muito difícil resolver independência sem utilizar funções de suavização para tratar instabilidades numéricas quando o número de termos na série é muito grande. Em tais casos, é necessário achar uma função de densidade apropriada para descrever os dados, tornando a solução mais complexa.

Em nossa abordagem, nós relaxamos a restrição de independência entre todos os classificadores binários. Para tal, nós achamos grupos de classificadores afins. Dentro de um grupo, há grande dependência entre os classificadores, enquanto que cada grupo é independente dos outros. No entanto, como a hipótese de independência é apenas entre os grupos, há menor possibilidade de incorrer em instabilidade numérica ou utilizar funções de suavização.

Nós utilizamos o conjunto de dados de treinamento para achar as probabilidades conjuntas das dicotomias dentro de um grupo e construir a respectiva tabela de probabilidade condicional (CPT) para este grupo de dicotomias afins.

Nós modelamos o problema de classificação multi-classe condicionado a grupos de dicotomias afins $\mathcal{G}_\mathcal{D}$. O modelo na Equação 1.4 torna-se

$$P(y = c_i|\mathcal{O}_\mathcal{T}, \mathcal{G}_\mathcal{D}) \propto P(\mathcal{O}_\mathcal{T}, \mathcal{G}_\mathcal{D}|y = c_i)P(y = c_i). \tag{1.7}$$

Nós assumimos independência apenas entre os grupos de dicotomias afins $g_i \in \mathcal{G}_\mathcal{D}$. Desta forma, a classe de uma entrada x é dada por

$$cl(x) = \arg\max_j \prod_{g_i \,\in\, \mathcal{G}_\mathcal{D}} P(\mathcal{O}^{g_i}_\mathcal{T}, g_i|y = c_j)P(y = c_j). \tag{1.8}$$

Para achar os grupos de classificadores binários afins $\mathcal{G}_{\mathcal{D}}$, nós definimos uma matriz de afinidade $\mathcal{A}$ entre os classificadores. Esta matriz indica quão afins (correlacionadas) são duas dicotomias quando classificando um conjunto de dados de treinamento X. Se as dicotomias produzem saídas todas iguais (diferentes), elas são correlacionadas e tem alta afinidade. Por outro lado, se seus resultados são metade iguais e metade diferentes, elas são não-correlacionadas e, portanto, possuem baixa afinidade.

Após o cálculo da matriz de afinidade $\mathcal{A}$, nós utilizamos um algoritmo de clusterização para achar grupos de classificadores binários afins em $\mathcal{A}$. Os grupos de classificadores afins podem conter classificadores que não contribuem muito para o processo geral de classificação. No processo de *Shrinking*, apresentamos um procedimento para identificar as dicotomias menos importantes dentro de um grupo de classificadores binários afins e eliminá-los. Para isso, calculamos a entropia acumulada de cada grupo testando um elemento do grupo de cada vez. Aqueles que produzem o menor ganho de informação são marcados como menos importantes.

A eliminação de dicotomias menos importantes nos abre a oportunidade de substituí-las por outras mais discriminativas. No processo de *Augmenting*, encontramos novas dicotomias candidatas para repor aquelas eliminadas na etapa de *Shrinking*. Para isso, analisamos a matriz de confusão calculada durante o treinamento. Em seguida, representamos as classes como um grafo onde os nós são os identificadores das classes e as arestas o grau de confusão. A partir do grafo, conseguimos criar uma hierarquia de classes em confusão. Após ordenarmos os grupos de classes de acordo com a sua confusão, achamos o corte de cada subgrafo que nos permite separar otimamente os nós. Isso nos dá conjuntos de dicotomias que representam classes em confusão e podem ser substitutas daquelas eliminadas no processo de *Shrinking*.

Finalmente, podemos utilizar as etapas de *Shrinking* e *Augmenting* iterativamente de modo a otimizar ainda mais o algoritmo base do *Affine-Bayes*.

1.4.2 Resultados obtidos

O Capítulo 6 é uma compilação de nosso trabalho submetido à *IEEE Transactions on Pattern Analysis and Machine Intelligence* (TPAMI) e do artigo [150] no *Intl. Conference on Computer Vision Theory and Applications* (VISAPP).

Detecção de Adulterações em Imagens Digitais

No Capítulo 2, apresentamos um estudo crítico das principais técnicas existentes na análise forense de imagens. Discutimos que boa parte dos problemas relacionados à análise forense apontam para dois pontos em comum: a seleção de características e as técnicas de aprendizado a serem utilizadas.

Conforme argumentamos, ainda não existem metodologias estabelecidas para verificar a autenticidade e integridade de imagens digitais de forma automática.

A identificação de imagens que foram digitalmente adulteradas é de fundamental importância atualmente [43,138,141]. O julgamento de um crime, por exemplo, pode estar sendo baseado em evidências que foram fabricadas especificamente para enganar e mudar a opinião de um júri. Um político pode ter a opinião pública lançada contra ele por ter aparecido ao lado de um traficante procurado mesmo sem nunca ter visto este traficante antes. Dessa forma, discutimos também questões legais associadas à análise forense de imagens como, por exemplo, o uso de fotografias digitais por criminosos.

O trabalho apresentado no Capítulo 2 é uma compilação de nosso artigo submetido ao *ACM Computing Surveys*. Os autores desse artigo, em ordem, são: Anderson Rocha, Walter J. Scheirer, Terrance E. Boult e Siome Goldenstein.

O banco de dados de imagens que discutimos nesse capítulo é resultado de nosso artigo [154] no *IEEE Workshop on Vision of the Unseen* (WVU).

Chapter 2

Current Trends and Challenges in Digital Image Forensics

Abstract

Digital images are everywhere — from our cell phones to the pages of our newspapers. How we choose to use digital image processing raises a surprising host of legal and ethical questions we must address. What are the ramifications of hiding data within an innocent image? Is this security when used legitimately, or intentional deception? Is tampering with an image appropriate in cases where the image might affect public behavior? Does an image represent a crime, or is it simply a representation of a scene that has never existed? Before action can even be taken on the basis of a questionable image, we must detect something about the image itself. Investigators from a diverse set of fields require the best possible tools to tackle the challenges presented by the malicious use of today's digital image processing techniques.

In this paper, we introduce the emerging field of digital image forensics, including the main topic areas of source camera identification, forgery detection, and steganalysis. In source camera identification, we seek to identify the particular model of a camera, or the exact camera, that produced an image. Forgery detection's goal is to establish the authenticity of an image, or to expose any potential tampering the image might have undergone. With steganalysis, the detection of hidden data within an image is performed, with a possible attempt to recover any detected data. Each of these components of digital image forensics is described in detail, along with a critical analysis of the state of the art, and recommendations for the direction of future research.

2.1 Introduction

With the advent of the Internet and low-price digital cameras, as well as powerful image edition software tools (Adobe Photoshop and Illustrator, GNU Gimp), normal users have become digital doctoring specialists. At the same time our understanding of the technological, ethical, and legal implications associated with image editing falls far behind. When such modifications are no longer innocent image tinkerings and start implying legal threats to a society, it becomes paramount to devise and deploy efficient and effective approaches to detect such activities [141].

Digital Image and Video Forensics research aims at uncovering and analyzing the underlying facts about an image/video. Its main objectives comprise: tampering detection (cloning, healing, retouching, splicing), hidden data detection/recovery, and source identification with no prior measurement or registration of the image (the availability of the original reference image or video).

Image doctoring in order to represent a scene that never happened is as old as the art of the photograph itself. Shortly after the Frenchman Nicéphore Niepce [29] created the first photograph in 1814[1], there were the first indications of doctored photographs. Figure 2.1 depicts one of the first examples of image forgery. The photograph, an analog composition comprising 30 images[2], is known as *The Two Ways of Life* and was created by Oscar G. Rejland in 1857.

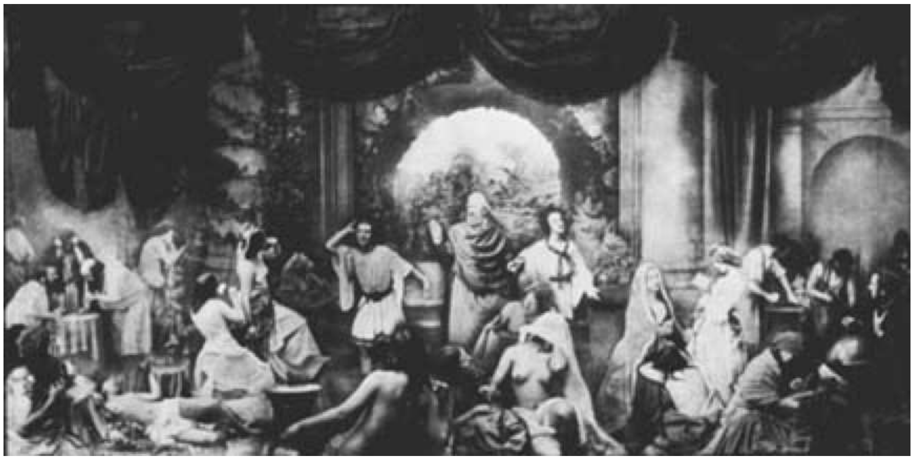


Figure 2.1: Oscar Rejland's analog composition, 1857.

Though image manipulation is not new, its prevalence in criminal activity has surged over the past two decades, as the necessary tools have become more readily available, and easier to use. In the criminal justice arena, we most often find tampered images in connection with child pornography cases. The 1996 Child Pornography Prevention Act (CPPA) extended the existing federal criminal laws against child pornography to include certain types of "virtual porn". Notwithstanding, in 2002, the United States Supreme Court found that portions of the CPPA, being excessively broad and restrictive, violated First Amendment rights. The Court ruled that images containing an actual minor or portions of a minor are not protected, while computer generated images depicting a fictitious "computer generated" minor are constitutionally protected. However, with computer graphics, it is possible to create fake scenes visually indistinguishable from real ones. In this sense, one can apply sophisticated approaches to give

[1] Recent studies [101] have pointed out that the photograph was, indeed, invented concurrently by several researchers such as Nicéphore Niepce, Louis Daguerre, Fox Talbot, and Hercule Florence.

[2] Available in `http://www.bradley.edu/exhibit96/about/twoways.html`

more realism to the created scenes deceiving the casual eye and conveying a criminal activity. In the United States, a legal burden exists to "a strong showing of the photograph's competency and authenticity[3]" when such evidence is presented in court. In response, tampering detection and source identification are tools to satisfy this requirement.

Data hidden within digital imagery represents a new opportunity for classic crimes. Most notably, the investigation of Juan Carlos Ramirez Abadia, a Columbian drug trafficker arrested in Brazil in 2008, uncovered voice and text messages hidden within images of a popular cartoon character[4] on the suspect's computer. Similarly, a 2007 study[5] performed by Purdue University found data hiding tools on numerous computers seized in conjunction with child pornography and financial fraud cases. While a serious hinderance to a criminal investigation, data hiding is not a crime in itself; crimes can be masked by its use. Thus, an investigator's goal here is to identify and recover any hidden evidence within suspect imagery.

In our digital age, images and videos fly to us at remarkable speed and frequency. Unfortunately, there are currently no established methodologies to verify their authenticity and integrity in an automatic manner. Digital image and video forensics are still emerging research fields with important implications for ensuring the credibility of digital contents. As a consequence, on a daily basis we are faced with numerous images and videos — and it is likely that at least a few have undergone some level of manipulation. The implications of such tampering are only beginning to be understood.

Beyond crime, the scientific community has also been subject to these forgeries. A recent case of scientific fraud involving doctored images in a renowned scientific publication has shed light to a problem believed to be far from the academy. In 2004, the South Korean professor Hwang Woo-Suk and colleagues published in *Science* important results regarding advances in stem cell research. Less than one year later, an investigative panel pointed out that nine out of eleven customized stem cell colonies that Hwang had claimed to have made involved doctored photographs of two other, authentic, colonies. Sadly, this is not a detached case. In at least one journal[6] [129], it is estimated that as many as 20% of the accepted manuscripts contain figures with improper manipulations, and +1% with fraudulent manipulations [45, 129].

Photo and video retouching and manipulation are also present in general press media. On July 10^{th}, 2008, various major daily newspapers published a photograph of four Iranian missiles streaking heavenward (see Figure 2.2(a)). Surprisingly, shortly after the photo's publication, a small blog provided evidence that the photograph had been doctored. Many of those same newspapers needed to publish a plethora of retractions and apologies [107].

On March 31st, 2003 the Los Angeles Times showed on its front cover an image from photojournalist Brian Walki, in which a British soldier in Iraq stood trying to control a crowd of civilians in a passionate manner. The problem was that the moment depicted never happened (see Figure 2.2(b)). The photograph was a composite of two different photographs merged

[3] Bergner v. State, 397 N.E.2d 1012, 1016 (Ind. Ct. App. 1979).

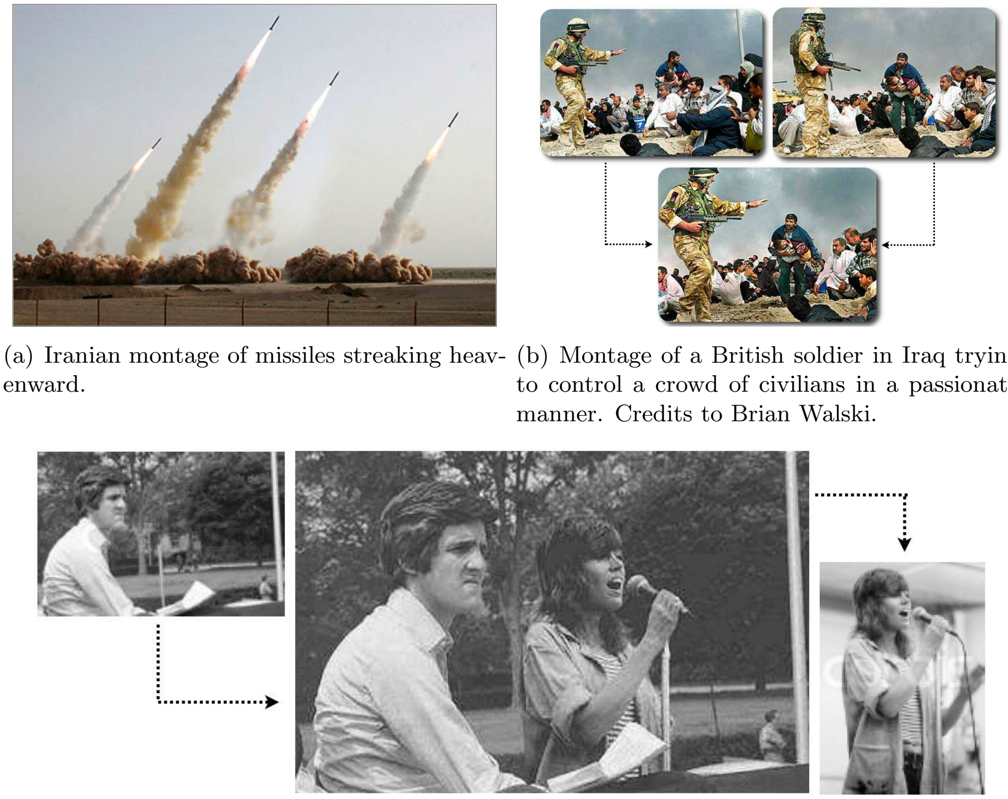
[4] http://afp.google.com/article/ALeqM5ieuIvbrvmfofmOt8oOYfXzbysVuQ

[5] http://www.darkreading.com/security/encryption/showArticle.jhtml?articleID=208804788

[6] Journal of Cell Biology.

to create a more appealing image. The doctoring was discovered and Walski was fired.

In the 2004 presidential campaign, John Kerry's allies were surprised by a photomontage that appeared in several newspapers purporting to show Kerry and Jane Fonda standing together at a podium during a 1970s anti-war rally (see Figure 2.2(c)). As a matter of fact, the photograph was a fake. Kerry's picture was taken at an anti-war rally in Mineola, NY., on June 13th, 1971 by photographer Ken Light. Fonda's picture was taken during a speech at Miami Beach, FL. in August, 1972 by photographer Owen Franken.



(a) Iranian montage of missiles streaking heavenward.

(b) Montage of a British soldier in Iraq trying to control a crowd of civilians in a passionate manner. Credits to Brian Walski.

(c) Montage of John Kerry and Jane Fonda standing together at a podium during a 1970s anti-war rally. Credits to Ken Light (left), AP Photo (middle), and Owen Franken (right).

Figure 2.2: Some common press media photomontages.

It has long been said that an image worth a thousand words. Recently, a study conducted by Italian Psychologists have investigated how doctored photographs of past public events affect memory of those events. Their results indicate that doctored photographs of past public events can influence memory, attitudes and behavioral intentions [158]. That might be one of the reasons that several dictatorial regimes used to wipe out of their photographic records images of people who had fallen out of favor with the system [44].

In the following sections, we provide a comprehensive survey of the most relevant works with respect to this exciting new field of the *unseen* in digital imagery. We emphasize approaches that we believe to be more applicable to forensics. Notwithstanding, most publications in this emerging field still lack important discussions about resilience to counter-attacks, which anticipate the existence of forensic techniques [58]. As a result, the question of trustworthiness of digital forensics arises, for which we try to provide some positive insights.

2.2 Vision techniques for the Unseen

In this section, we survey several state-of-the-art approaches for image and video forgery detection, pointing out their advantages and limitations.

2.2.1 Image manipulation techniques

In the forensic point of view, it is paramount to distinguish simple image enhancements from image doctoring. Although there is a thin edge separating both, in the following we try to make this distinction clear.

On one extreme, we define image enhancements as operations performed in one image with the intention to improve its visibility. There is no local manipulation or pixel combination. Some image operations in this category are contrast and brightness adjustments, gamma correction, scaling, and rotation, among others. On the other extreme, image tampering operations are those with the intention to deceive the viewer at some level. In these operations, normally one performs localized image operations such as pixel combinations and tweaks, copy/paste, and composition with other images. In between these extremes, there are some image operations that by themselves are not considered forgery creation operations but might be combined for such objective. Image sharpening, blurring, and compression are some of such operations.

Some common image manipulations with the intention of deceiving a viewer:

1. **Composition or splicing**. It consists in the composition (merging) of an image I_c using parts of one or more parts of images $I_1 \ldots I_k$. For example, with this approach, a politician in I_1 can be merged beside a person from I_2, without even knowing such person.

2. **Retouching, healing, cloning**. These approaches consist in the alteration of parts of an image or video using parts or properties of the same image or video. Using such techniques, one can make a person 10 or 20 years younger (retouching and healing) or even change a crime scene eliminating a person in a photograph (cloning).

3. **Content embedding or Steganography**. It consists in the alteration of statistical or structural properties of images and videos in order to embed hidden contents. Most of the changes are not visually detectable.

Figure 2.3 depicts some possible image manipulations. From the original image (top left), we clone several small parts of the same image in order to eliminate some parts of it (for example, the two people standing in front of the hills). Then we can use a process of smoothing to feather edges and make the cloning less noticeable. We can use this image as a host for another image (bottom left) and then create a composite. After the combination, we can use healing operations to adjust brightness, contrast, and illumination. This toy example was created in five minutes using the open-source software Gimp.

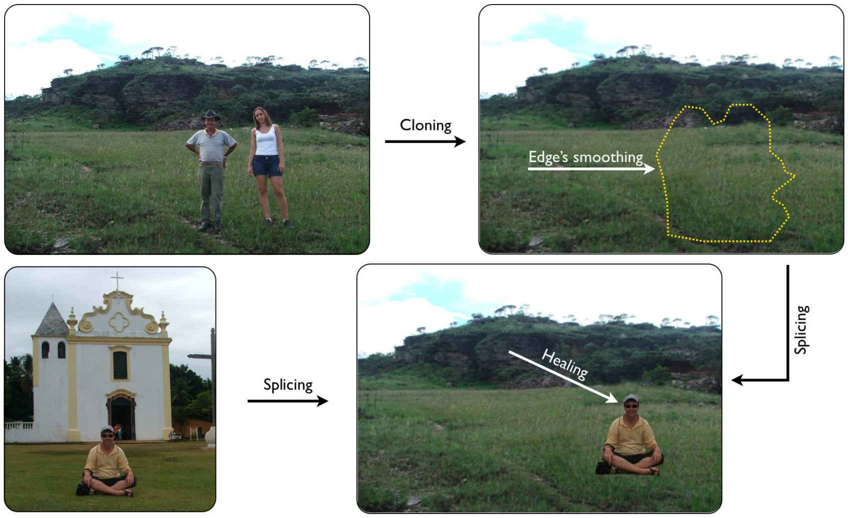


Figure 2.3: Toy example of possible image manipulations.

Sometimes the edge between image enhancing and faking is so thin that depending on the context, only the addition of text to a scene may fool the viewer. Figure 2.4 depicts one example of two photographs presented by Colin Powell at the United Nations in 2003. The actual photographs are low-resolution, muddy aerial surveillance photographs of buildings and vehicles on the ground in Iraq. They were used to justify a war. Note that the text addition in this case was enough to mislead the United Nations [104].

2.2.2 Important questions

In general, in digital image and video forensics, given an input digital image, for instance, one wants to answer the following important questions [161]:

- Is this image an original image or has it been created from the composition (splicing) of other images?
- Does this image represent a real moment in time or has it been tampered with to deceive the viewer?
- What is the processing history of this image?
- Which part of this image has undergone manipulation and to what extent? What are the impacts of such modifications?
- Was this image acquired from camera vendor X or Y?

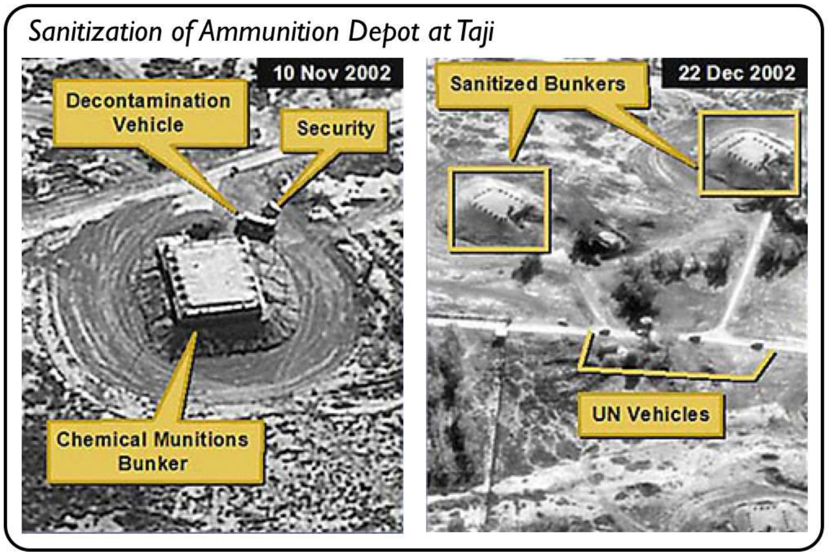


Figure 2.4: Photographs presented by Colin Powell at the United Nations in 2003. (U.S. Department of State)

- Was this image originally acquired with camera C as claimed?
- Does this image conceal any hidden content? Which algorithm or software has been used to perform the hiding? Is it possible to recover the hidden content?

It is worth noting that most of such techniques are blind and passive. The approach is blind when it does not use the original content for the analysis. The approach is passive when it does not use any watermarking-based solution for the analysis.

Although digital watermarking can be used in some situations, the vast majority of digital contents do not have any digital marking. Any watermarking-based solution would require an implementation directly in the acquisition sensor, making its use restrictive. Furthermore, such approaches might lead to quality loss due to the markings [118, 161].

We break up the image and video forensics approaches proposed in the literature in three categories:

1. Camera sensor fingerprinting or source identification;
2. Image and video tampering detection;
3. Image and video hidden content detection/recovery.

2.2.3 Source Camera Identification

With *Source Camera Identification*, we are interested in identifying the data acquisition device that generated a given image for forensics purposes. Source camera identification may be broken into two classes: device class identification and specific device identification. In general,

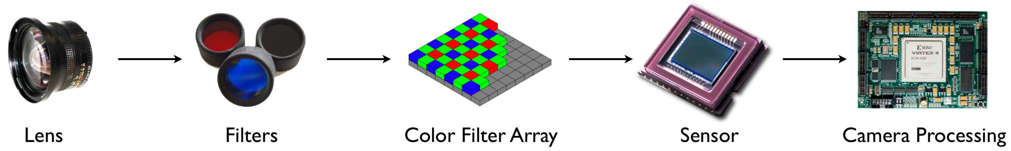


Figure 2.5: The image acquisition pipeline.

source camera identification relies on the underlying characteristics of the components of digital cameras. These characteristics may take the form of image artifacts, distortions, and statistical properties of the underlying data. These characteristics are usually imperceptible to the human eye, but visible effects can also contribute clues for identification.

In general, we treat digital image acquisition as a pipeline of stages. Figure 2.5 illustrates the flow of data, with light initially passing through a lens and possibly through a filter (to remove infrared or ultra-violet light, for example). If the camera supports color, a Color Filter Array (CFA) is usually placed over the sensor to accommodate different color channels. Popular CFA configurations include the RGB Bayer Pattern (most common), and the CMYK subtractive color model (available on some higher end sensors). In a standard consumer grade camera, the sensor will be a silicon CCD or CMOS. The image processing will take place in logic designed by individual camera or chipset manufacturers within the camera itself. Each of these pipeline components induce anomalies in images that can be used to identify a source camera.

Device Class Identification

The goal of device class identification is to identify the model and/or manufacturer of the device that produced the image in question. For digital cameras, we consider the image acquisition pipeline, where the lens, size of the sensor, choice of CFA, and demosaicing and color processing algorithms found in the camera processing logic to provide features. It is important to note that many manufacturers use the same components, thus, the discriminatory power of some techniques may be limited. Many of the techniques that we will discuss here treat the underlying camera characteristics as features for machine learning, which separates images into particular camera classes. Thus, we can treat device class identification as a traditional classification problem. Support Vector Machines (SVM), shown in Figure 2.6, is a popular binary classifier for device class separation. It can also be extended for multi-class classification. In this section, we will review the relevant techniques used to identify device classes.

From the lens, radial distortions can be introduced immediately into the image acquisition pipeline. Radial distortion is commonly found with inexpensive cameras/lenses. Choi et al. [27] introduces a method to extract aberrations from images, which are then treated as features for classification. As described in [27], radial distortion can be modeled through the second order

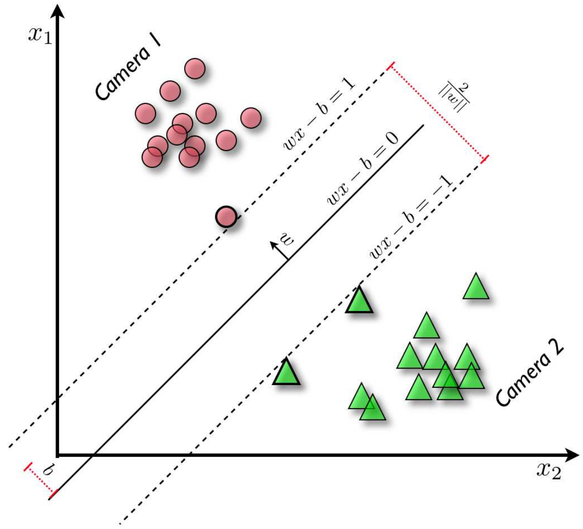


Figure 2.6: An example of binary camera classification with SVM. A feature vector is constructed out of the calculated features for a given image. Training sets are built out of a collection of feature vectors for each camera class. The machine learning is used for classification of images with unknown sources.

for reasonable accuracy:

$$r_u = r_d + k_1 r_d^3 + k_2 r_d^5 \tag{2.1}$$

where k_1 and k_2 are the first and second degree distortion parameters, and r_u and r_d are the undistorted radius and the distorted radius. The radius is simply the radial distance $\sqrt{x^2 + y^2}$ of some point (x, y) from the center of the distortion (typically the center of the image). k_1 and k_2 are treated as features for an SVM learning system. These features, however, are not used in [27] by themselves — they are combined with the 34 image features introduced in [81] (described below), in a fusion approach. Thus, the utility of this approach may be seen as a supplement to other, stronger features derived from elsewhere in the acquisition pipeline. The average accuracy of this technique is reported to be about 91% for experiments performed on three different cameras from different manufacturers.

Image color features exist as artifacts induced by the CFA and demosaicing algorithm of a color camera, and represent a rich feature set for machine learning based classification. Karrazi et al. [81] defines a set of image color features that are shown to be accurate for device class identification using SVMs. Average pixel values, RGB pairs correlation, neighbor distribution center of mass, RGB pairs energy ratio, and wavelet domain statistics are all used as features. Further, image quality features are also used to supplement the color features in [81]. Pixel difference based measures (including mean square error, mean absolute error, and modified infinity norm), correlation based measures (including normalized cross correlation, and the Czekonowksi correlation, described below), and spectral distance based measures (including spectral phase and magnitude errors) are all used. For binary classification, Kharrazi et al. [81] reports between 90.74% and 96.08% prediction accuracy. For multi-classification considering 5 cameras, prediction accuracy between 78.71% and 95.24% is reported. These results were confirmed in [176].

The CFA itself as a provider of features for classification has been studied in [23]. The motivation for using just the CFA and its associated demosaicing algorithm is that proprietary demosaicing algorithms leave correlations across adjacent bit planes of the images. Celiktutan et al. [23] defines a set of similarity measures $\{m_1, m_2, m_3\}$, with kNN and SVM used for classification.

The first approach is a binary similarity measure. A stencil function is first defined:

$$\alpha_c^n(k,b) = \begin{bmatrix} 1 & if & x_c = 0 & x_n = 0 \\ 2 & if & x_c = 0 & x_n = 1 \\ 3 & if & x_c = 1 & x_n = 0 \\ 1 & if & x_c = 1 & x_n = 1 \end{bmatrix} \tag{2.2}$$

where b is a bit plane (image matrix), the subscript c defines some central pixel, and n denotes one of the four possible neighbor pixels. The function is summed over its four neighbors, as well as all of the pixels in the bit plane. k indicates one of four agreement scores: 1,2,3,4. $\alpha_c^n(k,b)$ is summed over its four neighbors, and over all MxN pixels. Before feature generation, the agreement scores are normalized:

$$p_k^b = \alpha(k,b) / \sum_k (k,b) \tag{2.3}$$

p is the normalized agreement score in the Kullback-Leibler distance, m_1 defined as:

$$m_1 = -\sum_{n=1}^{4} p_n^7 \log \frac{p_n^7}{p_n^8} \tag{2.4}$$

The second approach is also a binary similarity measure, but uses a neighborhood weighting mask as opposed to a stencil function. Each binary image yields a 512-bin histogram computed using the weighted neighborhood. Each score is computed with the following function:

$$S = \sum_{i=0}^{7} x_i 2^i \tag{2.5}$$

The neighborhood weighting mask applied to a pixel x_i by the above function is:

1	2	4
128	256	8
64	32	16

The final binary similarity is computed based on the absolute difference between the n^{th} histogram bin in the 7^{th} bit plane and same of the 8^{th} after normalization:

$$m_2 = \sum_{n=0}^{511} |S_n^7 - S_n^8| \tag{2.6}$$

Quality measures, as mentioned earlier, make excellent features for classification. The Czenakowski distance is a popular feature for CFA identification because it is able to compare vectors with non-negative components — exactly what we find in color images. The third feature of [23] is the Czenakowski distance defined as:

$$m_3 = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(1 - \frac{2 \sum_{k=1}^{3} min(C_k(i,j), \hat{C}_k(i,j))}{\sum_{k=1}^{3} (C_k(i,j) + \hat{C}_k(i,j))} \right) \tag{2.7}$$

Denoising is necessary for calculating this distance metric. $C_k(i,j)$ represents the $(i,j)^{th}$ pixel of the k^{th} band of a color image, with $\hat{C}_k$ being the denoised version. With these three similarity measures the authors of [23] generate 108 binary similarity features and 10 image quality similarity features per image. The best reported performance for this technique (using SVM for classification) is near 100% accuracy for the two camera classification problem, 95% accuracy for the three camera classification problem, and 62.3% accuracy for a six camera classification problem.

A major weakness of the approaches described thus far is a lack of rigor in the analysis of the experimental results reported, compared with other security related vision and pattern recognition fields such as biometrics and tracking. All report raw classification results for only a handful of different cameras. Thus, it is often difficult to determine how well these techniques perform in practice. This is a common problem of this sub-field in general. By varying the SVM margin after classification, a set of marginal distances can be used to build a Receiver Operator Characteristic curve. From this curve, a more thorough understanding of the False Reject Rate (FRR) and False Accept Rate (FAR) can be gained. Also of interest is more comprehensive testing beyond limited camera classes. For a more accurate picture of the FAR, a statistically large sampling of images from cameras outside the known camera classes should be submitted to a system. None of the papers surveyed attempted this experiment. Further, the techniques introduced thus far are all shown to succeed on images with low levels of JPEG compression. How well these techniques work with high levels of compression has yet to be shown. Not all work suffers from a dearth of analysis, however.

The Expectation/Maximization algorithm [138] is a powerful technique for identifying demosaicing algorithms, and does not rely on classification techniques directly, but can take advantage of them in extensions to the base work ([12] , [13]). The motivating assumption of the E/M algorithm is that rows and columns of interpolated images are likely to be correlated with their neighbors. Kernels of a specified size (3 × 3, 4 × 4, and 5 × 5 are popular choices) provide this neighborhood information to the algorithm. The algorithm itself can be broken into two steps. In the *Expectation* step, the probability of each sample belonging to a particular model is estimated. In the *Maximization* step, the specific form of the correlations between samples is estimated. Both steps are iterated till convergence.

In detail, we can assume that each sample belongs to one of two models. If a sample is linearly correlated with its neighbors, it belongs to M_1. If a sample is not correlated with its neighbors, it belongs to M_2. The linear correlation function is defined as:

$$f(x,y) = \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v) + n(x,y) \tag{2.8}$$

In this linear model, $f(\cdot,\cdot)$ is a color channel (R, G, or B) from a demosaiced image, N is an integer, and $n(x,y)$ represents independent, identically distributed samples drawn from a Gaussian distribution with zero mean and unknown variance. $\vec{\alpha}$ is a vector of linear coefficients that express the correlations, with $\alpha_{0,0} = 0$.

The expectation step estimates the probability of each sample belonging to M_1 using Bayes' rule:

$$\Pr\{f(x,y) \in M_1 | f(x,y)\} = \frac{\Pr\{f(x,y)|f(x,y) \in M_1\}\Pr\{f(x,y) \in M_1\}}{\sum_{i=1}^{2} \Pr\{f(x,y)|f(x,y) \in M_i\}\Pr\{f(x,y) \in M_i\}} \tag{2.9}$$

$\Pr\{f(x,y) \in M_1\}$ and $\Pr\{f(x,y) \in M_2\}$ are prior probabilities assumed to be equal to 1/2. If we assume a sample $f(x,y)$ is generated by M_1, the probability of this is:

$$\Pr\{f(x,y)|f(x,y) \in M_1\} = \frac{1}{\sigma\sqrt{2\pi}}\left[-\frac{1}{2\sigma^2}\left(f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u,y+v)\right)^2\right]. \tag{2.10}$$

We estimate the variance σ^2 in the Maximization step. M_2 is assumed to have a uniform distribution.

The Maximization step computes an estimate of $\vec{\alpha}$ using weighted least squares (in the first round of the Expectation step, $\vec{\alpha}$ is chosen randomly):

$$E(\vec{\alpha} = \sum_{x,y} w(x,y)\left(f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u,y+v)\right)^2 \tag{2.11}$$

The weights $w(x,y)$ are equivalent to $\Pr\{f(x,y) \in M_1 | f(x,y)\}$. This error function is minimized via a system of linear equations before yielding its estimate. Both the steps are executed until a stable $\vec{\alpha}$ results. The final result maximizes the likelihood of observed samples.

Popescu [138] asserts that the probability maps generated by the E/M algorithm can be used to determine which demosaicing algorithm a particular camera is using. These probabilities tend to cluster — thus, an external machine learning algorithm for classification is not necessary. For a test using eight different demosaicing algorithms [138], the E/M algorithm achieves an average classification accuracy of 97%. In the worst case presented (3×3 median filter vs. variable number of gradients), the algorithm achieves an accuracy of 87%. Several extensions to the E/M algorithm have been proposed. Bayram et al. [12] applies the E/M algorithm to a camera identification problem, using SVM to classify the probability maps. Bayram et al. [12] reports success as high as 96.43% for the binary classification problem, and 89.28% for the multi-class problem. Bayram et al. [13] introduces better detection of interpolation artifacts in smooth images as a feature to fuse with the standard E/M results. For a three camera identification problem, Bayram et al. [13] achieves results as high as 97.74% classification accuracy. Other variations include the use of modeling error, instead of interpolation filter coefficients [89], and the computation of error based on the assumption of CFA patterns in an image [172].

Specific Device Identification

The goal of specific device identification is to identify the exact device that produced the image in question. For specific device identification, we require more detail beyond what we've discussed so far with source model identification. Features in this case may be derived from:

- hardware and component imperfections, defects, and faults
- effects of manufacturing process, environment, operating conditions
- aberrations produced by a lens, noisy sensor, dust on the lens

It is important to note that these artifacts may be temporal by nature, and thus, not reliable in certain circumstances.

Early work [76] in imaging sensor imperfections for specific device identification focused on detecting fixed pattern noise caused by *dark current* in digital video cameras. Dark current is the rate that electrons accumulate in each pixel due to thermal action. This thermal energy is found within inverse pin junctions of the sensor, and is independent of light falling on it. The work, as presented in [76], provides no quantitative analysis, and thus, the actual utility of dark currents cannot be assessed.

A more comprehensive use of sensor imperfections is presented in [57], where "hot pixels," cold/dead pixels, pixel traps, and cluster defects are used for detection. Hot pixels are individual pixels on the sensor with higher than normal charge leakage. Cold or dead pixels (figure 2.7) are pixels where no charge ever registers. Pixel traps are an interference with the charge transfer process and results in either a partial or whole bad line, that is either all white or all dark. While these features are compelling for identifying an individual sensor, Geradts et at. [57] also does not provide a quantitative analysis. Thus, we turn to more extensive work for reliable forensics.

Lukas et al. [91] presents a more formal quantification and analysis of sensor noise for identification, with work that is the strongest for this type of forensics. Referring to the hierarchy of sensor noise in Figure 2.8, we see two main types of pattern noise: fixed pattern noise and photo-response non-uniformity noise. Fixed pattern noise (FPN) is caused by the dark currents described above, and is not considered in [91]. Photo-response non-uniformity noise (PRNU) is primarily cause by pixel non-uniformity noise (PNU). PNU is defined as different sensitivity various pixels have to light caused by the inconsistencies of the sensor manufacturing process. Low frequency defects are caused by light refraction on particles on or near the camera, optical surfaces, and zoom settings. Lukas et al. [91] does not consider this type of noise, but [39] does.

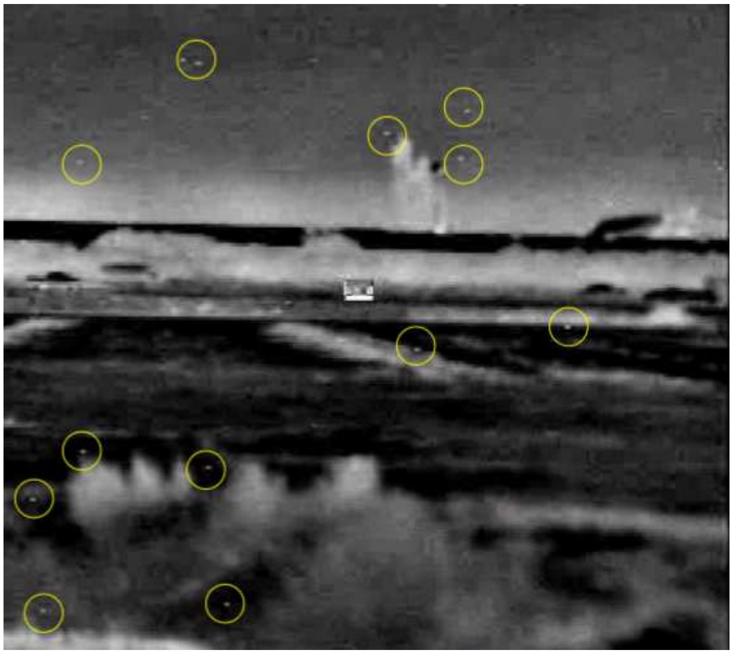


Figure 2.7: Dead pixels (circled in yellow) present in an image from a thermal surveillance camera.

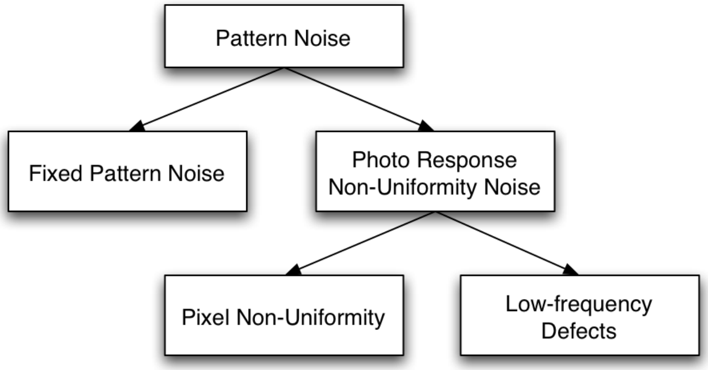


Figure 2.8: Hierarchy of Pattern Noise.

The temporal nature of such particle artifacts brings into question their reliability, except when dealing with short sequences of images from the same period, in most cases.

To use PNU as a characteristic for sensor fingerprinting, the nature of the noise must first be isolated. An image signal r exhibits properties of a white noise signal with an attenuated high frequency band. The attenuation is attributed to the low-pass character of the CFA algorithm (which, in this case, we are not interested in). If a large portion of the image is saturated (pixel values set to 255), it will not be possible to separate the PNU from the image signal. In a forensics scenario, we will likely not have a blank reference image that will easily allow us to gather the PNU characteristics. Thus, the first stage of the PNU camera identification algorithm is to establish a reference pattern P_c, which is an approximation to the PNU. The approximation, $\bar{p}^{(k)}$ is built from the average of N different images:

$$\bar{p}^{(k)} = \frac{1}{N}\sum_{i=1}^{N} p^i \tag{2.12}$$

The approximation can be optimized to suppress the scene content by applying a de-noising filter F, and averaging the noise residuals $n^{(k)}$ instead of the original images $P^{(k)}$:

$$\bar{n}^{(k)} = (\bar{p}^{(k)} - F(p^{(k)}))/N \tag{2.13}$$

Lukas et al. [91] reports that a wavelet-based denoising filter works the best.

To determine if an image belongs to a particular known camera, a correlation ρ_c is simply calculated between the noise residual of the image in question $n = p - F(p)$ and the reference pattern P_c:

$$\rho_c(p) = \frac{(n - \bar{n}) \cdot (P_c - \bar{P}_c)}{\|n - \bar{n}\| \|P_c - \bar{P}_c\|} \tag{2.14}$$

The results of [91] are expressed in terms of FRR and FAR (proper ROC curves are not provided, however), with very low FRR (between 5.75×10^{-11} and 1.87×10^{-3}) reported when a FAR of 10^{-3} is set for an experiment with images from nine different cameras. Excellent correlations are shown for all tests, indicating the power this technique has for digital image forensics. An enhancement to this work has been proposed by [171], with a technique to fuse the demosaicing characteristics of a camera described earlier with the PNU noise. Performance is enhanced by as much as 17% in that work over the base PNU classification accuracy.

Counter Forensic Techniques Against Camera Identification

Like any sub-field of digital forensics, camera identification is susceptible to counter forensic techniques. Gloe et al. [58] introduces two techniques for manipulating the image source identification of [91]. This work makes the observation that applying the wavelet denoising filter of [91] is not sufficient for creating a quality image. Thus, a different method, *flatfielding*, is applied to estimate the FPN and the PRNU. FPN is a signal *independent* additive noise source, while PRNU is a signal *dependent* multiplicative noise source. For the FPN estimate, a dark frame d is created by averaging K images x_{dark} taken in the dark (with the lens cap on, for instance):

$$d = \frac{1}{K}\sum_{K} x_{dark} \tag{2.15}$$

For the PRNU estimate, L images of a homogeneously illuminated scene x_{light} with d subtracted are required. To calculate the flatfield frame f, these images are averaged:

$$f = \frac{1}{L}\sum_{L}(x_{light} - d) \tag{2.16}$$

With an estimate of the FPN and PRNU of a camera, a nefarious individual can suppress the noise characteristics of an image from a particular camera to avoid identification. An image $\hat{x}$ with suppressed noise characteristics is simply created by noise minimization:

$$\hat{x} = \frac{x - d}{f} \tag{2.17}$$

The authors of [58] note that perfect flatfielding is, of course, not achievable, as an immense number of parameters (exposure time, shutter speed, and ISO speed) would be needed to generate d and f. Thus, they fix upon a single parameter set for their experiments. Results for this technique are reported for RAW and TIFF images. While powerful, flatfielding is not able to prevent identification in all images it is applied to.

Simply reducing the impact of camera identification by PRNU is not the only thing one can do with flatfielding. After the above technique has been applied, a noise pattern from a different camera can be added with inverse flatfielding. An image $\hat{y}$ with forged noise characteristics is created from the pre-computed flatfielding information from any desired camera:

$$\hat{y} = \hat{x} \cdot f_{forge} + d_{forge} \tag{2.18}$$

Experiments for this technique are also presented in [58], where images from a Canon Powershot S70 are altered to appear to be from a Canon Powershot S45. While most correlation coefficients mimic the S45, some still remain characteristic of the S70. The counter forensic techniques of [58] are indeed powerful, but are shown to be too simplistic to fool a detection system absolutely. Further, such limited testing only hints at the potential of such techniques. As the “arms race” continues, we expect attacks against camera identification to increase in sophistication, allowing for more comprehensive parameter coverage and better noise modeling.

2.2.4 Image and video tampering detection

In general, image and video tampering detection approaches rely on analyzing several properties such as: detection of cloned regions, analysis of features’ variations collected from sets of original and tampered scenes, inconsistencies in the features, inconsistencies regarding the acquisition process, or even structural inconsistencies present in targeted attacks. In the following, we describe each one of such approaches and their limitations.

Image cloning detection

Cloning is one of the simplest forgeries an image can undergo. It is known as copy/move and also is present in more sophisticated operations such as healing. Often, the objective of the cloning operation is to make an object “disappear” from one scene using properties of the same scene (for example, neighboring pixels with similar properties). Cloning detection is a problem technically easy to solve using exhaustive search. However, brute-force solutions are computationally expensive.

Fridrich et al. [53] have proposed a faster and more robust approach for detecting image duplicated regions in images. The authors use a sliding window over the image and calculate the discrete cosine transform (DCT) for each region. Each calculated DCT window is stored row-wise in a matrix A. The authors propose to calculate a quantized DCT in order to be more robust and perform matchings for non-exact cloned regions. The next step consists of lexicographically sorting matrix A and searching for similar rows. To reduce the resulting false positives, the authors proposed a post-processing step in which they only consider two rows as a clone candidate if more rows share the same condition and are close in the image space to these two rows. Popescu and Farid [139] proposed a similar approach switching the DCT calculation to a Karhunen-Loeve Transform and reported comparable results.

As we discussed in Section 2.1, forgeries are also present in the scientific community. Some authors may use image tampering to improve their results and make them look more attractive. Farid [45] has framed the detection of some scientific image manipulations as a two-stage segmentation problem. The proposed solution is suited for grayscale images such as gel DNA response maps. In the first iteration, the image is grouped, using intensity-based segmentation into regions corresponding to the bands (gray pixels) and the background. In the second iteration, the background region is further grouped into two regions (black and white pixels) using the texture-based segmentation. Both segmentations are performed using Normalized cuts [165]. The authors suggest that the healing and cloning operations will result in large segmented cohesive regions in the background that are detectable using a sliding window and ad-hoc thresholds. This approach seems to work well for naive healing and cloning operations, but only a few images were tested. It would be interesting to verify if a copied band of another image still would lead to the same artifacts when spliced in the host image.

Video splicing and cloning detection

Wang and Farid [187] have argued that the two previous approaches are too computationally inefficient to be used in videos or even for small sequences of frames and proposed an alternative solution to detect duplicated regions across frames. Given a pair of frames $f(x, y, \tau_1)$ and $f(x, y, \tau_2)$, from a stationary camera, the objective is to estimate a spatial offset (Δ_x, Δ_y) corresponding to a duplicated region of one frame placed in another frame in a different spatial location. Towards this objective, the authors use phase correlation estimation [22]. First, the normalized cross power spectrum is defined:

$$P(\omega_x, \omega_y) = \frac{F(\omega_x, \omega_y, \tau_1)F^*(\omega_x, \omega_y, \tau_2)}{||F(\omega_x, \omega_y, \tau_1)F^*(\omega_x, \omega_y, \tau_2||}, \tag{2.19}$$

where $F(\cdot)$ is the Fourier transform of a frame, $*$ is the complex conjugate, and $||\cdot||$ is the complex magnitude. Phase correlation techniques estimate spatial offsets by extracting peaks in $p(x, y)$, the inverse Fourier transform of $P(\omega_x, \omega_y)$. A peak is expected at origin (0,0) as it is a stationary camera. Peaks at other positions denote secondary alignments that may represent a duplication but also simple camera translations (for non-stationary cameras). The spatial location of a peak corresponds to candidate spatial offsets (Δ_x, Δ_y). For each spatial offset, the authors calculate the correlation between $f(x, y, \tau_1)$ and $f(x, y, \tau_2)$ to determine if an offset corresponds to a determined duplication. Toward this objective, each frame is tiled into 16×16 overlapping (1 pixel) blocks and the correlation coefficient between each pair of corresponding blocks is computed. Blocks whose correlation is above a threshold are flagged as duplications. The authors also propose an extension for non-stationary cameras. For that, they calculate a rough measure of the camera motion and compensate the calculation by selecting subsequent non-overlapping frames. One drawback of this approach is that it assumes that the duplicated regions are rough operations (do not undergo significant adjustments in the host frame).

Wang and Farid [186] presented an approach for detecting traces of tampering in interlaced and de-interlaced videos. For de-interlaced videos, the authors use an expectation maximization algorithm to estimate the parameters of the underlying de-interlacing algorithm. With this model, the authors can point out the spatial/temporal correlations. Tampering in the video is likely to leave telltale artifacts that disturb the spatial/temporal correlations. For interlaced videos, the authors measure the inter-field and inter-frame motion which are often the same for an authentic video, but may be different for a doctored video. Although effective to some extent, it is worth discussing some possible limitations. The solution suitable for interlaced videos is sensitive to compression artifacts hardening the correlations estimation. In addition, a counter-attack to the de-interlacing approach consists of performing the video tampering and then generating an interlaced video (splitting the even and odd scan lines), and applying a de-interlacing algorithm on top of that to generate a new de-interlaced video whose correlations will be intact.

Variations in image features

Avcibas et al. [11] have framed the image forgery detection problem as a feature and classification fusion problem. The authors claim that doctoring typically involves multiple steps, which often demand a sequence of elementary image processing operations such as scaling, rotation, contrast shift, smoothing, among others. The authors develop single weak "experts" to detect each such elementary operations. Thereafter, these weak classifiers are fused. The authors have used features borrowed from the Steganalysis literature (c.f., Sec. 2.2.5) such as image quality metrics [10], binary similarity measures [8], and high order separable quadrature mirror filters statistics [97]. The main limitation with such approach is that the elementary operations by themselves do not constitute doctoring operations. Hence, this approach needs to be used wisely to point out localized operations. In this case, abrupt brightness and contrast changes in regions in the host image may point to forgeries (for example, when splicing different images). However, local intrinsic changes need to be accounted for in order to reduce the high rate of false positives. Finally, for criminal forgeries, it is likely that the forger will seek to match the target and host images in such a way to reduce these subtleties.

Ng and Chang [117] have proposed a feature-based binary classification system using high order statistics to detect image composition. For that, the authors use bicoherence features motivated by the effectiveness of the bicoherence features for human-speech splicing detection [116]. Bicoherence is the third order correlation of three harmonically related Fourier frequencies of a signal $X(\omega)$ (normalized bispectrum). The authors report an accuracy of $\approx 71\%$ on the Columbia Splicing data set. The Columbia data set, however, is composed of small composite images without any kind of post-processing. Figure 2.9 depicts four images in such a data set. Finally, it is worth noting that the bicoherence features calculation is a computational intensive procedure, often $O(N^4)$ where N is the number of pixels of an image.

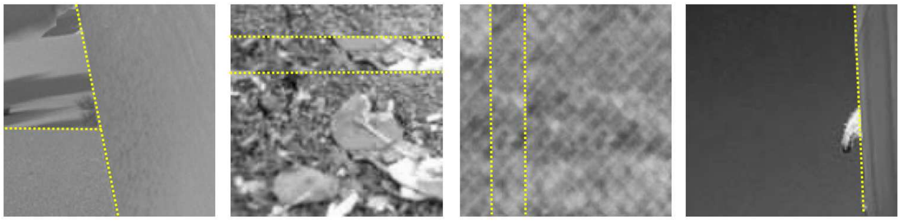


Figure 2.9: Some examples from the Columbia Splicing data set. We emphasize the splicing boundaries in yellow.

Shi et al. [167] have proposed a natural image model to separate spliced images from natural images. The model is represented by features extracted from a given set of test images and 2-D arrays produced by applying multi-size block discrete cosine transform (MBCT) to the given image. For each 2-D array, the authors calculate a prediction-error 2-D array, its wavelet subbands, and 1-D and 2-D statistical moments. In addition, the authors also calculate Markov transition probability matrices for the 2-D arrays differences which are taken as additional features. Although effective for simple image splicing procedures (copying and pasting) such as the ones in the Columbia Splicing data set [103] with $\approx 92\%$ accuracy, the approach does not seem to be effective for more sophisticated compositions that deploy adaptive edges and structural propagation [170]. This is because the transition matrices often are unable to capture the subtle edge variation upon structural propagation. In addition, such an approach is a binary-based solution. Up to now, it does not point out possible forgery candidate regions.

Inconsistencies in image features

When splicing two images to create a composite, one often needs to re-sample an image onto a new sampling lattice using an interpolation technique (such as bi-cubic). Although imperceptible, the re-sampling contains specific correlations that, when detected, may represent evidence of tampering. Popescu and Farid [141] have described the form of these correlations, and proposed an algorithm for detecting them in an image. The authors showed that the specific form of the correlations can be determined by finding the neighborhood size, N, and the set of coefficients, $\vec{\alpha}$, that satisfy: $\vec{a}_i = \sum_{k=-N}^{N} \alpha_k \vec{a}_{i+k}$ in the equation

$$\left(\vec{a}_i - \sum_{k=-N}^{N} \alpha_k \vec{a}_{i+k} \right) \cdot \vec{x} = 0, \tag{2.20}$$

where $\vec{x}$ is the signal, and $\vec{a}_i$ is the i^{th} row of the re-sampled matrix. The authors pointed out that, in practice, neither the samples that are correlated, nor the specific form of the correlations are known. Therefore, the authors employ an expectation maximization algorithm (EM) similar to the one in Section 2.2.3 to simultaneously estimate a set of periodic samples correlated to their neighbors and, an approximation form for these correlations. The authors assume that each sample belongs to one of two models. The first model M_1, corresponds to those samples y_i that are correlated to their neighbors and are generated according to the following model:

$$M_1 : y_1 = \sum_{k}^{-N} N \alpha_k y_{i+k} + n(i), \tag{2.21}$$

where $n(i)$ denote independently, and identically distributed samples drawn from a Gaussian distribution with zero mean an unknown variance σ^2. In the E-step, the probability that each sample y_i belonging to model M_1 can be estimated through Bayes rule similarly to the Equation 2.9, Section 2.2.3, where y_i replaces $f(x,y)$. The probability of observing a sample y_i knowing it was generated by M_1 is calculated in the same way as in Equation 2.10, Section 2.2.3, where y_i replaces $f(x,y)$. The authors claim that the generalization of their algorithm to color images is fairly straightforward. The authors propose to analyze each color channel independently. However, the authors do not show experiments for the performance of their algorithm under such circumstances and to what extent such independence assumption is valid. Given that demosaiced color images present high pixel correlation, such analysis would be valuable.

It is assumed that the probability of observing samples generated by the outlier model, $Pr\{y_i | y_i \in M_2\}$, is uniformly distributed over the range of possible values of y_i. Although, it might seem a strong assumption, the authors do not go into more detail justifying the choice of the uniform distribution for this particular problem. In the M-step, the specific form of the correlations between samples is estimated minimizing a quadratic error function. It is important to note that the re-sampling itself does not constitute tampering. One could just save space by down-sampling every picture in a collection of pictures. However, when different correlations are present in one image, there is a strong indication of image composition. The authors have reported very good results for high-quality images. As the image is compressed, specially under JPEG 2000, the re-sampling correlates and hence tampering becomes harder to detect. It is worth noting that it is also possible to perform a counter attack anticipating the tampering detection and, therefore, destroying traces of re-sampling. Gloe et al. [58] presented a targeted attack in which the pixel correlations are destroyed by small controlled geometric distortions. The authors superimpose a random disturbance vector $\vec{e}$ to each individual pixel's position. To deal with possible jitter effects, the strength of distortion is adaptively modulated by the local image content using simple edge detectors.

When creating a digital composite (for example, two people standing together), it is often difficult to match the lighting conditions from the individual photographs. Johnson and Farid [70] have presented a solution that analyzes lighting inconsistencies to reveal traces of digital tampering. Standard approaches for estimating light source direction begin by making some simplifying assumptions such as: (1) the surface is Lambertian (it reflects light isotropically); (2) it has a constant reflectance value; (3) it is illuminated by a point light source infinitely far away; among others. However, to estimate the lighting direction, standard solutions require knowledge of the 3-D surface normals from, at least, four distinct points on a surface with same reflectance, which is hard to find from a single image and no objects of known geometry in the scene. The authors have used a clever solution first proposed by [121] that estimates two components of the light source direction from a single image. The authors also relax the constant reflectance assumption by assuming that the reflectance for a local surface patch is constant. This requires the technique to estimate individual light source directions for each patch along a surface. Figure 2.10(a) depicts an example where lighting inconsistencies can point out traces of tampering.

More recently, Johnson and Farid [71] have extended this solution to complex lighting environments by using spherical harmonics. Under the aforementioned simplifying assumptions, an arbitrary lighting environment can be expressed as a non-negative function on the sphere, $L(\vec{V})$, $\vec{V}$ is a unit vector in Cartesian coordinates and the value of $L(\vec{V})$ is the intensity of the incident light along direction $\vec{V}$. If the object being illuminated is convex, the irradiance (light received) at any point on the surface is due to only lighting environment (no cast shadows or inter-reflections).

It is worth noting, however, that even if the authors' assumptions were true for an object, which is rather limiting, they are virtually never true for a scene of interest given that a collection of convex objects is no longer convex.

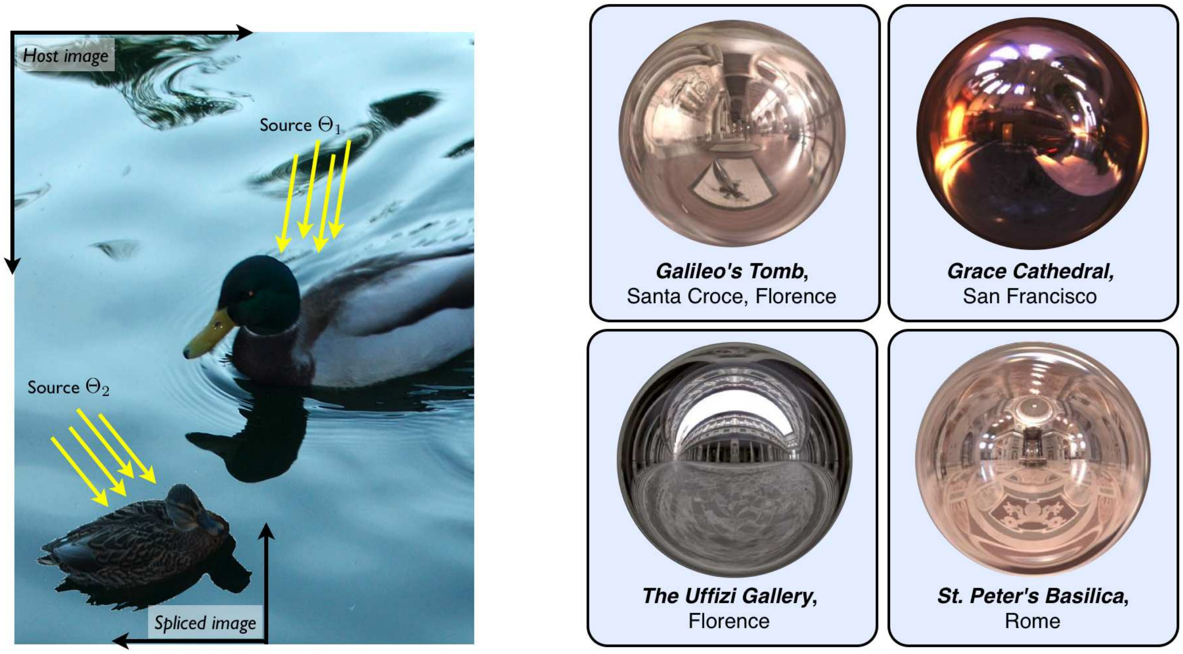
As a result, the irradiance, $E(\vec{N})$, can be parametrized by the unit length surface normal $\vec{N}$ and written as a convolution of the reflectance function on the surface, $R(\vec{V}, \vec{N})$, with the lighting environment $L(\vec{V})$:

$$E(\vec{N}) = \int_{\Omega} L(\vec{V}) R(\vec{V}, \vec{N}) d\Omega \tag{2.22}$$

where Ω represents the surface. For a Lambertian surface, the reflectance function is a clamped cosine:

$$R(\vec{V}, \vec{N}) = \max(\vec{V} \cdot \vec{N}, 0). \tag{2.23}$$

The convolution in Equation 2.22 can be simplified by expressing both the lighting environment and the reflectance functions in terms of spherical harmonics. The main drawback of such an approach is that in order to generate a good estimation of the lighting environment it is necessary to learn the light behavior on a series of light probe images. A light probe image is an omnidirectional, high dynamic range image that records the incident illumination conditions at a particular point in space (see Figure 2.10(b)). Lighting environments can be captured by a variety of methods such as photographing a mirror sphere or through panoramic photographic techniques [36]. This is necessary to represent the lighting environment function $L(\vec{V})$ that is then integrated to result in the spherical harmonics representing the scene lighting.



(a) Composite example with lighting inconsistencies.

(b) Four light probes from different lighting environments. Credits to Paul Debevec and Dan Lemmon.

Figure 2.10: Lighting and forgeries.

More recently, Johnson and Farid [72] have also investigated lighting inconsistencies across specular highlights on the eyes to identify composites of people. The position of a specular highlight is determined by the relative positions of the light source, the reflective surface and the viewer (or camera). According to the authors, specular highlights that appear on the eye are a powerful cue as to the shape, color, and location of the light source(s). Inconsistencies in these properties of the light can be used as telltales of tampering. It is worth noting that specular highlights tend to be relatively small on the eye giving room to a more skilled forger to manipulate them to conceal traces of tampering. To do so, shape, color, and location of the highlight would have to be constructed so as to be globally consistent with the lighting in other parts of the image.

Acquisition inconsistencies

In the same way that we can use camera properties to point out the sensor that captured an image, we also can use them as a digital X-ray for revealing forgeries [25].

Lin et al. [87] have presented an approach that explores camera response normality and consistency functions to find tampering footprints. An image is tagged as doctored if the response functions are abnormal or inconsistent to each other. The camera response function is a mapping relationship between the pixel irradiance and the pixel value. For instance, suppose a pixel is on an edge and the scene radiance changes across the edge and is constant on both sides of the edge (Figure 2.11(a)). Therefore, the irradiance of the pixel on the edge should be a linear combination of those of the pixels clear off the edges (Figure 2.11(b)). Due to nonlinear response of the camera, the linear relationship breaks up among the read-out values of these pixels (Figure 2.11(c)). The authors estimate the original linear relationship when calculating the inverse camera response function [86]. Although effective in some situations, this approach

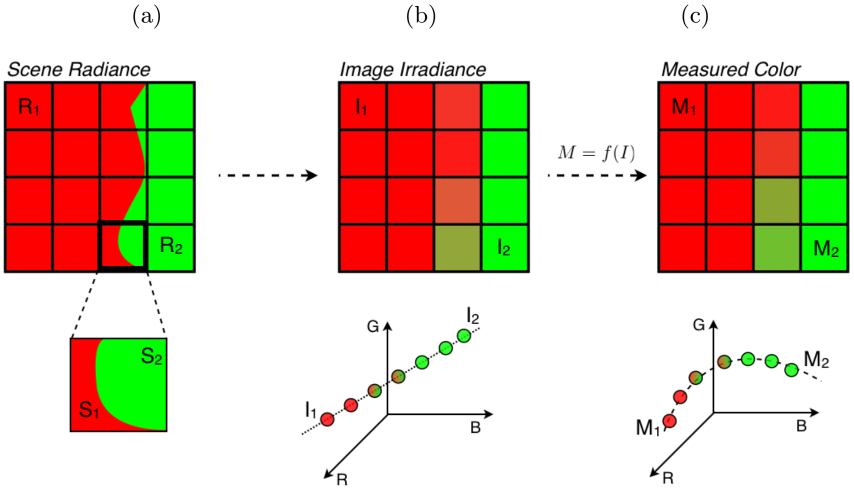


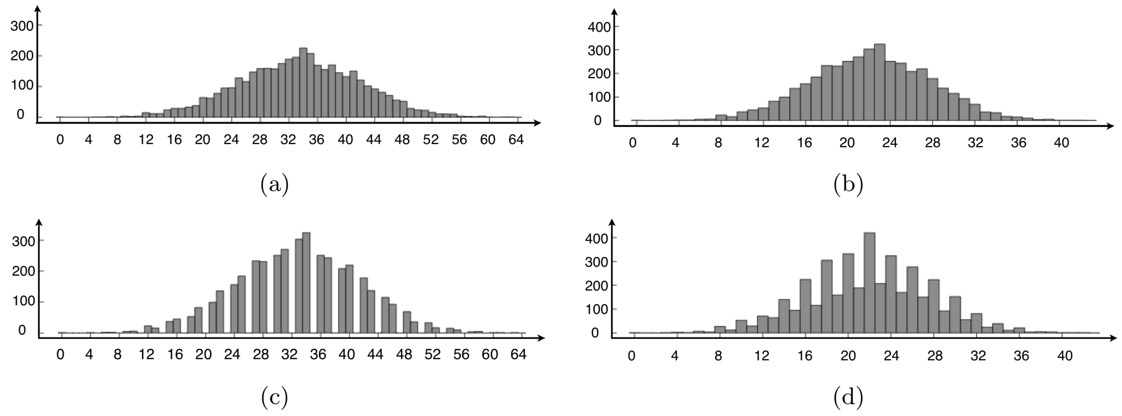
Figure 2.11: Camera Response Function Estimation. (a) R_1 and R_2 are two regions with constant radiance. The third column images are a combination of R_1 and R_2. (b) The irradiances of pixels in R_1 map to the same point I_1, in RGB color space. The same happens for pixels in R_2 which maps to I_2. However, the colors of the pixels in the third column is the linear combination of I_1 and I_2. (c) The camera response function f warps the line segment in (b) into a curve during read-out.

has several drawbacks. Namely, (1) to estimate the camera response function, the authors must calculate an inverse camera response function which requires learning a Gaussian Mixture Model from a database with several known camera response functions (DoRF) [87]. If the analyzed image is a composite of regions from unknown cameras, the model is unable to point out an estimation for the camera response function; (2) the approach requires the user to manually select points on edges believed to be candidates for splicing; (3) the solution requires high contrast images to perform accurate edge and camera normality estimations; (4) the approach might fail if the spliced images are captured by the same camera and not synthesized along the edges of an object; (5) Finally, it is likely the solution does not work with CMOS adaptive sensors that dynamically calculate the camera response function to produce more pleasing pictures.

Chen et al. [25] have proposed to use inconsistencies in the photo-response non-uniformity noise (c.f., Sec. 2.2.3) to detect traces of tampering. The method assumes that either the camera that took the image or at least some other pristine images taken by the camera are available. The algorithm starts by sliding a 128 × 128 block across the image and calculating the value of the test statistics, $p_{\mathcal{B}}$, for each block $\mathcal{B}$. The probability distribution function $p(x|H_0)$ of $p_{\mathcal{B}}$ under H_0 is estimated by correlating the PRNU noise residuals from other cameras and is modeled as a generalized Gaussian. For each block, the pdf $p(x|H_1)$ is obtained from a block correlation predictor and is also modeled as a generalized Gaussian. For each block $\mathcal{B}$, the authors perform a Neyman-Pearson hypothesis testing by fixing the false alarm rate α and decide that $\mathcal{B}$ has been tampered if $p_{\mathcal{B}} < Th$. The threshold Th is determined from the condition $\alpha = \int_{Th} p(x|H_0)dx$.

Structural inconsistencies

Some forgery detection approaches are devised specifically for a target. Popescu and Farid [140] have discussed the effects of double quantization for JPEG images and presented a solution to detect such effects. Double JPEG compression introduces specific artifacts not present in single compressed images. The authors also note that evidence of double JPEG compression, however, does not necessarily prove malicious tampering. For example, it is possible for a user to simply re-save a high quality JPEG image with a lower quality. Figure 2.2.4 depicts an example of the double quantization effect over a 1-d toy example signal $x[t]$ normally distributed in the range $[0, 127]$.



(a)

(b)

(c)

(d)

Figure 2.12: The top row depicts histograms of single quantized signals with steps 2 (left) and 3 (right). The bottom row depicts histograms of double quantized signals with steps 3 followed by 2 (left), and 2 followed by 3 (right). Note the periodic artifacts in the histograms of double quantized signals. Credits to Alin Popescu.

Inspired by the pioneering work of [140] regarding double quantization effects and their use in forensics, He et al. [65] have proposed an approach to locate doctored parts in JPEG images by examining the double quantization effect hidden among DCT coefficients. The idea is that as long as a JPEG image contains both the doctored part and the pristine part, the discrete cosine coefficient histograms of the pristine part will still have the double quantization effect (DQ), because this part of the image is the same as that of the double compressed original JPEG image. However, the histograms of a doctored part will not have the same DQ effects. Some possible reasons for these observations are: (1) absence of the first JPEG compression in the doctored part; (2) mismatch of the DCT grid of the doctored part with that of the pristine part; or (3) composition of DCT blocks along the boundary may carry traces of the doctored and pristine parts given that it is not likely that the doctored part exactly consists of 8×8 blocks. It is worth noting, however, that this solution will not work in some circumstances. For instance, if the original image to contribute to the pristine part is not a JPEG image, the double quantization effect of the pristine part cannot be detected. In addition, the compression levels also affect the detection. Roughly speaking, the smaller the ratio of the second quantization step with respect to the first one, the harder the detection of the DQ effects. Finally, if the forger re-samples the grid of the DCT (shift the image one pixel), it is possible to destroy the traces of the double quantization and generate a complete new quantization table.

2.2.5 Image and video hidden content detection/recovery

Steganography is the art of secret communication. Its purpose is to hide the presence of communication — a very different goal than *Cryptography*, which aims to make communication unintelligible for those that do not possess the correct access rights [6].

Applications of Steganography can include feature location (identification of subcomponents within a data set), captioning, time-stamping, and tamper-proofing (demonstration that original contents have not been altered). Unfortunately, not all applications are harmless, and there are strong indications that Steganography has been used to spread child pornography on the Internet [64, 113], and as an advanced communication tool for terrorists and drug-dealers [105, 106].

In response to such problems, the forensic analysis of such systems is paramount. We refer to *Forensic Steganalysis* as the area related to the detection and recovery of hidden messages. In this forensic scenario, we want to distinguish *non-stego* or *cover objects*, those that do not contain a hidden message, and *stego-objects*, those that contain a hidden message with the additional requirement of recovering its content as a possible proof basis for the court.

Steganography and Steganalysis have received a lot of attention around the world in the past few years [149]. Some are interested in securing their communications through hiding the very fact that they are exchanging information. On the other hand, others are interested in detecting the existence of these communications — possibly because they might be related to illegal activities. In the aftermath of 9/11 events, some researchers have suggested that Osama Bin Laden and Al Qaeda used Steganography techniques to coordinate the World Trade Center attacks. Almost six years later, nothing was proved [21,83,149,184]. However, since then, there has been strong evidences that Steganography has been used as a private communication means for drug-dealers and child pornographers in their illegal activities [64, 105, 106, 113]. Indeed, according to the *High Technology Crimes Annual Report* [108,120], Steganography threats can also appear in conjunction with dozens of other cyber-crimes such as: fraud and theft, terrorism, computer cracking, online defamation, intellectual property offenses, and online harassment.

In the following sections, we present representative research with respect to the identification and recovery of hidden messages in digital multimedia. When possible, we emphasize approaches that can be used as an aid for criminal prosecution in a court of law. The fundamental goal of Steganalysis is to reliably detect the existence of hidden messages in communications and, indeed, most of the approaches in the literature have addressed only the detection problem. However, for forensics purposes, we are interested in the higher level of analysis going one step further and attempting to recover the hidden content.

We can model the detection of hidden messages in a cover medium as a classification problem. In Steganalysis, we have two extreme scenarios: (1) Eve has only some level of suspicion that Alice and Bob are covertly communicating; and (2) Eve may have some additional information about Alice and Bob's covert communications such as the algorithm they have used, for instance. In the first case, we have a difficult forensic scenario where Eve would need to deploy a system able to detect all forms of Steganography (*Blind Steganalysis*). In the latter case, Eve might have additional information reducing her universe of possible hiding algorithms and cover media (*Targeted Steganalysis*).

In general, steganographic algorithms rely on the replacement of some component of a digital object with a pseudo-random secret message [6]. In digital images, common components used to conceal data are: (1) the least significant bits (LSBs); (2) DCT coefficients in JPEG-compressed images; and (3) areas with richness in details [32].

Figure 2.13 depicts a typical Steganography and Steganalysis scenario. When embedding a message in an image, one can take several steps in order to avoid message detection such as choosing an embedding key, compressing the message, and applying statistical profiling in the message and the cover media in order to minimize the amount of changes. On the other hand, in the Steganalysis scenario, we can try to point out the concealment whether making statistical analysis on the input image, or on the image and on a set of positive and negative training examples. If we have additional information, we can also use them in order to perform a targeted attack. In the following, we present some approaches used to detect such activities using either targeted or blind attacks.

Targeted Steganalysis

Some successful approaches for targeted Steganalysis proposed in the literature can estimate the embedding ratio or even reveal the secret message with the knowledge of the steganographic algorithm being very useful for forensics.

Basic LSB embedding can be reliably detected using the histogram attack as proposed

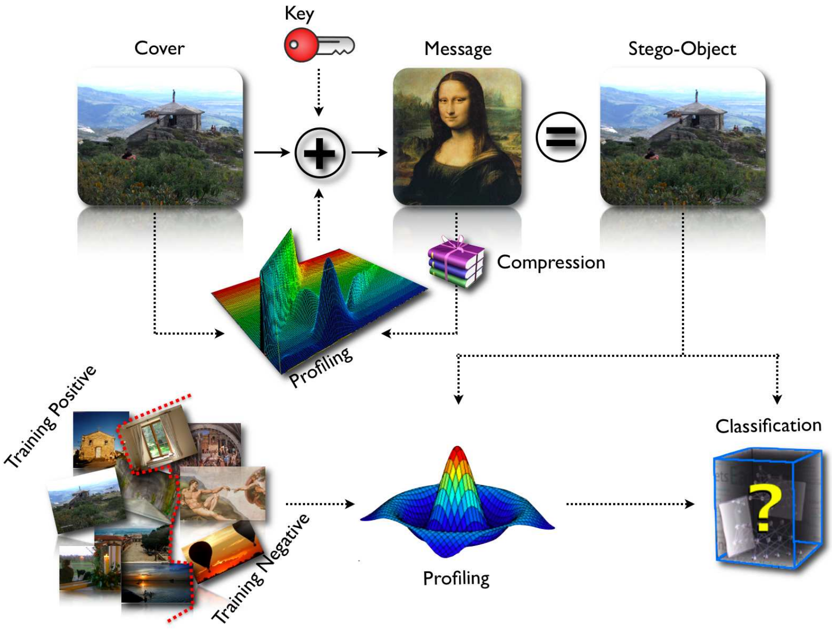


Figure 2.13: Typical Steganography and Steganalysis scenario.

by [191]. Any possible LSB embedding procedure will change the contents of a selected number of pixels and therefore will change the pixel value statistics in a local neighborhood.

An L-bit color channel can represent 2^L possible values. If we split these values into 2^{L-1} pairs that only differ in the LSBs, we are considering all possible patterns of neighboring bits for the LSBs. Each of these pairs are called *pair of value* (PoV) in the sequence [191].

When we use all the available LSB fields to hide a message in an image, the distribution of odd and even values of a PoV will be the same as the 0/1 distribution of the message bits. The idea of the statistical analysis is to compare the theoretically expected frequency distribution of the PoVs with the real observed ones [191]. However, we do not have the original image and thus the expected frequency. In the original image, the theoretically expected frequency is the arithmetical mean of the two frequencies in a PoV. As we know, the embedding function only affects the LSBs, so it does not affect the PoV's distribution after an embedding. Therefore the arithmetical mean remains the same in each PoV, and we can derive the expected frequency through the arithmetic mean between the two frequencies in each PoV.

As presented in [143,191], we can apply the χ^2 (chi squared-test) S over these PoVs to detect hidden messages

$$S = \sum_{i=1}^{k} \frac{(f_i^{obs} - f_i^{exp})^2}{f_i^{exp}}, \tag{2.24}$$

where k is the number of analyzed PoVs, f_i^{obs} and f_i^{exp} are the observed frequencies and the expected frequencies respectively. A small value of S points out that the data follows the expected distribution and we can conclude that the image was tweaked. We can measure the statistical significance of S by calculating the p-value, which is the probability that a chi-square distributed random variable with $k-1$ degrees of freedom would attain a value larger than or equal to S:

$$p(S) = \frac{1}{2^{\frac{k-1}{2}}\Gamma(\frac{k-1}{2})} \int_{S}^{\infty} e^{\frac{-x}{2}} x^{\frac{k-1}{2}-1} dx. \tag{2.25}$$

If the image does not have a hidden message, S is large and $p(S)$ is small. In practice, we calculate a threshold value S_{th} so that $p(S_{th}) = \alpha$ where α is the chosen significance level. The main limitation with this approach is that it only detects sequential embeddings. For random embeddings, we could apply this approach window-wise. However, in this case it is effective only for large embeddings such as the ones that modify, at least, 50% of the available LSBs. For small embeddings, there is a simple counter-attack that breaks down this detection technique. For that, it is possible to learn the basic statistics about the image and to keep such statistics when embedding the message. For instance, for each bit modified to one, another one is flipped to zero. Indeed, as we shall show later, Outguess[7] is one approach that uses such tricks when performing embeddings in digital images.

Fridrich et al. [50] have presented RS analysis. It consists of the analysis of the LSB loss-less embedding capacity in color and gray-scale images. The loss-less capacity reflects the fact that the LSB plane — even though it looks random — is related to the other bit planes [50]. Modifications in the LSB plane can lead to statistically detectable artifacts in the other bit planes of the image. The authors have reported good results (detection for message-sizes as small as $\approx 2-5\%$ on a limited set of images for the Steganography tools: Steganos, S-Tools, Hide4PGP, among others[8].

A similar approach was devised by [40] and is known as *sample pair analysis*. Such an approach relies on the formation of some subsets of pixels whose cardinalities change with LSB embedding, and such changes can be precisely quantified under the assumption that the embedded bits form a random walk on the image. Consider the partitioning of the input image in vectorized form V into pairs of pixels (u, v). Let $\mathcal{P}$ be the set of all pairs. Let us partition $\mathcal{P}$ into three disjoint sets X, Y, and Z, where

$$\begin{aligned} X &= \{(u,v) \in \mathcal{P} \quad | \quad (v \text{ is even and } u < v) \text{ or } (v \text{ is odd and } u > v)\ \} \\ Y &= \{(u,v) \in \mathcal{P} \quad | \quad (v \text{ is even and } u > v) \text{ or } (v \text{ is odd and } u < v)\ \} \\ Z &= \{(u,v) \in \mathcal{P} \quad | \quad (u = v)\} \end{aligned} \tag{2.26}$$

Furthermore, let us partition the subset Y into two subsets, W, and V, where $V = Y \setminus W$, and

$$Y = \{(u,v) \in \mathcal{P} \mid (u = 2k, v = 2k+1) \text{ or } (u = 2k+1, v = 2k)\} \tag{2.27}$$

The sets X, W, V, and Z are called primary sets and $\mathcal{P} = X \cup W \cup V \cup Z$. When one embeds content in an image, the LSB values are altered and therefore the cardinalities of the sets will change accordingly. As we show in Figure 2.14, we have four possible cases $\pi \in \{00, 01, 10, 11\}$.

[7] `http://www.outguess.org/`

[8] `http://members.tripod.com/steganography/stego/software.html`

Let p be the relative amount of modified pixels in one image due to embedding. Hence, the probability of a state change is given by

$$\begin{aligned}\rho(00,\mathcal{P}) &= (1-p/2)^2\\ \rho(01,\mathcal{P}) &= \rho(10,\mathcal{P}) = p/2(1-p/2)^2\\ \rho(11,\mathcal{P}) &= (p/2)^2.\end{aligned} \tag{2.28}$$

and the cardinalities after the changes are

$$\begin{aligned}|X'| &= |X|(1-p/2)+|V|p/2\\ |V'| &= |V|(1-p/2)+|X|p/2\\ |W'| &= |W|(1-p+p^2/2)+|Z|p(1-p/2)\end{aligned} \tag{2.29}$$

It follows that

$$|X'|-|V'| = (|X|-|V|)(1-p). \tag{2.30}$$

The authors have empirically noted the, on average, for natural images (no hidden content) $|X|=|Y|$. Therefore,

$$|X'|-|V'| = |W|(1-p). \tag{2.31}$$

Observe in Figure 2.14 that the embedding process does not alter $W\cup Z$. Hence, we define $\gamma=|W|+|Z|=|W'|+|Z'|$ yielding

$$|W'| = (|X'|-|V'|)(1-p)^2+\gamma p(1-p/2). \tag{2.32}$$

Given that $|X'|+|V'|+|W'|+|Z'|=|\mathcal{P}|$, we have the estimation of the embedded content size

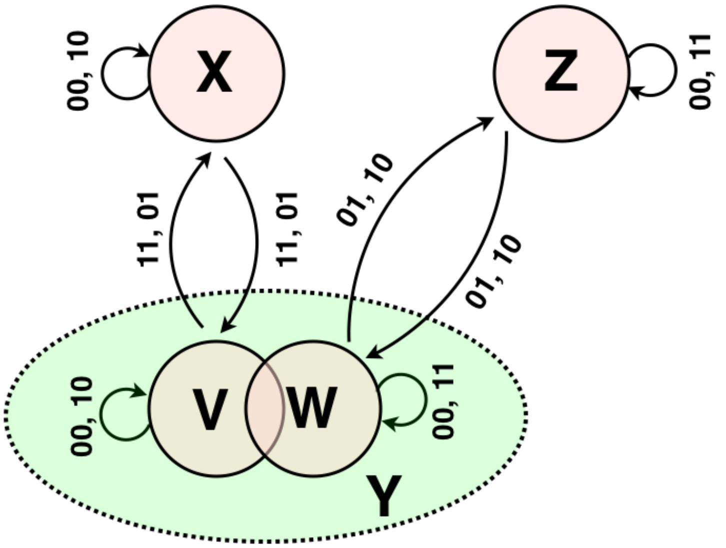


Figure 2.14: Transitions between primary sets under LSB changing.

$$0.5\gamma p^2 + (2|X'| - |\mathcal{P}|)p + |Y'| - |X'| = 0. \tag{2.33}$$

This approach has been tested in [32] over three data sets summing up to 5,000 images. The data sets comprise raw, compressed, and also scanned images. The approach is able to detect messages as small as 5% of the available space for normal LSB embedding with no statistical profiling.

Ker [80] has studied the statistical properties of the analysis of pairs and also proposed an extension using weighted least squares [80]. Recently, Bohme [17] presented an extension for JPEG covers. Several other approaches have been designed to detect targeted Steganalysis specifically in the JPEG domain [49, 56, 133].

Shi et al. [166] have analyzed the gradient energy flipping rate during the embedding process. The hypothesis is that the gradient energy varies consistently when the image is altered to conceal data.

For most of the above techniques, the authors do not discuss possible counter-attacks to their solutions. For instance, the sample pairs solution [40] and the RS analysis [50] rely on the analysis of groups of modified and non-modified pixels. What happens if someone knows these detection solutions and compensates for the group distribution for each modified pixel? Do the solutions still work after such kind of embedding statistical profiling?

Blind Steganalysis

Most of the blind- and semi-blind detection approaches rely on supervised learning techniques. The classifiers used in existing blind and semi-blind Steganalysis refer to virtually all categories of classical classification such as regression, multi-variate regression, one class, two class, and hyper-geometric classifications, among others.

Both in blind and semi-blind scenarios, the classifier is a mapping that depends on one or more parameters that are determined through training and based on the desired tradeoff between both type of errors (false alarm and false detection) that the classifier can make. Therefore, Steganalysis begins with the appropriate choice of features to represent both the stego and non-stego objects.

In the semi-blind scenario, we select a set of stego algorithms and train a classifier in the hope that when analyzing an object concealing a message embedded with an unknown algorithm, the detector will be able to generalize. On the other hand, in the complete blind scenario, we only train a set of cover objects based on features we believe will be altered during the concealment of data. In this case, we train one-class classifiers and use the trained model to detect outliers.

Some of the most common features used in the literature to feed classifiers are based on wavelet image decompositions, image quality metrics, controlled perturbations, moment functions, and histogram characteristic functions.

Lyu and Farid [95,96] have introduced a detection approach based on probability distribution functions of image sub-bands coefficients. This work has become a basis for several others. The motivation is that natural images have regularities that can be detected by high-order statistics through quadrature mirror filter (QMF) decompositions [180].

The QMF decomposition divides the image into multiple scales and orientations. We denote the vertical, horizontal, and diagonal sub-bands in a given scale $\{i = 1 \dots n\}$ as $V_i(x, y)$, $H_i(x, y)$, $D_i(x, y)$, respectively. Figure 2.15 depicts one image decomposition with three scales. The authors of [95, 96] propose to detect hidden messages using two sets of statistics collected throughout the multiple scales and orientations. The first set of statistics comprises *mean*, *variance*, *skewness*, and *kurtosis*. These statistics are unlikely to capture the strong correlations that exist across space, orientation, scale and color. Therefore, the authors calculate a second set of statistics based on the errors in a linear predictor of coefficient magnitude. For the sake of illustration, consider a vertical sub-band of a gray image at scale i, $V_i(x, y)$. A linear predictor for the magnitude of these coefficients in a subset of all possible spatial, orientation, and scale neighbors is given by

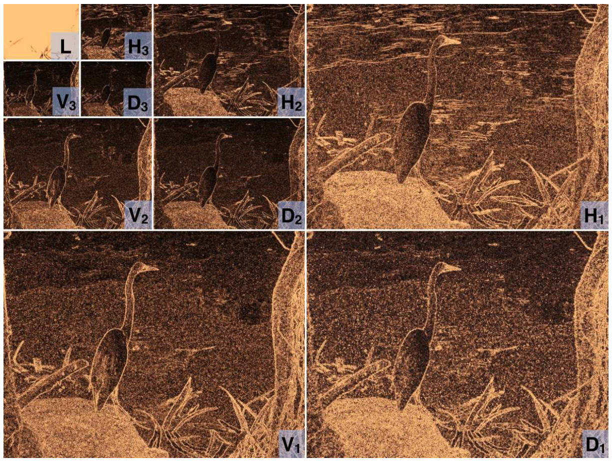


Figure 2.15: Image sub-bands QMF decomposition.

$$\begin{aligned} |V_i(x,y)| &= w_1|V_i(x-1,y)| + w_2|V_i(x+1,y)| + w_3|V_i(x,y-1)| + w_4|V_i(x,y+1)| \\ &+ w_5\left|V_{i+1}\left(\frac{x}{2},\frac{y}{2}\right)\right| + w_6|D_i(x,y)| + w_7\left|D_{i+1}\left(\frac{x}{2},\frac{y}{2}\right)\right|, \end{aligned} \quad (2.34)$$

where $|\cdot|$ represents absolute value and w_k are the weights. We can represent this linear relationship in matrix form as $\vec{V} = Q\vec{w}$, where the column vector $\vec{w} = (w_1, \dots, w_7)^T$, the vector $\vec{V}$ contains the coefficient magnitudes of $V_i(x, y)$ strung out into a column vector, and the columns of the matrix Q contain the neighboring coefficient magnitudes as in Equation 2.34 also strung out into column vectors. The coefficients are determined through the minimization of the quadratic error function

$$E(\vec{w}) = [\vec{V} - Q\vec{w}]^2. \quad (2.35)$$

This error is minimized through differentiation with respect to $\vec{w}$. Setting the result equal to zero, and solving for $\vec{w}$, we have

$$\vec{w} = (Q^{\mathrm{T}}Q)^{-1}Q^{\mathrm{T}}\vec{V}. \quad (2.36)$$

Finally, the log error in the linear predictor is given by

$$\vec{E} = log_2\vec{V} - log_2(Q\vec{w}). \quad (2.37)$$

It is from this error that the additional mean, variance, skewness, and kurtosis statistics are collected. This process is repeated for each sub-band, and scale. From this set of statistics, the authors train the detector with images with and without hidden messages.

Lyu and Farid [93, 97] have extended this set of features to color images and proposed an one-class classifier with hyper-spheres representing cover objects. Outliers of this model are tagged as stego objects. A similar procedure using Parzen-Windows was devised by [156] to detect anomalies in stego systems.

Rocha and Goldenstein [147] have presented the *Progressive Randomization* meta-descriptor for Steganalysis. The principle is that it captures the difference between image classes (with and without hidden messages) by analyzing the statistical artifacts inserted during controlled perturbation processes with increasing randomness.

Avcibas et al. [10] have presented a detection scheme based on image quality metrics (IQMs). The motivation is that the embedding can be understood as an addition of noise to the image therefore degrading its quality. They have used multivariate regression analysis. Avcibas et al. [8] have introduced an approach that explores binary similarity measures within image bit planes. The basic idea is that the correlation between the bit planes as well as the binary texture characteristics within the bit planes differ between a stego image and a cover image.

Histogram characteristic functions and statistics of empirical co-occurrence matrices also have been presented with relative success [26, 49, 168, 193, 194].

Despite of all the advances, one major drawback of the previous approaches is that most of them are only able to point out whether or not a given image contains a hidden message. Currently, with classifier-based blind or semi-blind approaches it is extremely difficult or even impossible to identify portions of the image where a message is hidden and perform message extraction or even only point out possible tools used in the embedding process. A second drawback in this body of work is the lack of counter-analysis techniques to assess the viability of the existing research. Outguess[9] [142] and F5 [190] are two early examples of such works.

Outguess is a steganographic algorithm that relies on data specific handlers that extract redundant bits and write them back after modification. For JPEG images, Outguess preserves statistics based on frequency counts. As a result, statistical tests based on simple frequency counts are unable to detect the presence of steganographic content [142]. Outguess uses a generic iterator object to select which bits in the data should be modified. In addition, F5 was proposed with the goal of providing high steganographic capacity without sacrificing security. Instead of LSB flipping (traditional embedding approaches), the embedding operation in F5 preserves the shape of the DCT histogram. The embedding is performed according to a pseudo-random path determined from a user pass-phrase. Later on, Fridrich et al. [52] have provided a targeted attack that detects embedded messages using F5 algorithm throughout a process called calibration. We estimate the original cover-object from the suspected stego-object. In the case of JPEG images, for instance, this is possible because the quantized DCT coefficients are robust to small distortions (the ones performed by some steganographic algorithms) [32].

[9] http://www.outguess.org/

Fridrich et al.'s [52] approach is no longer as effective if we improve F5 with some sort of statistical profiling preserving not only the DCT histogram shape but also compensating for the modified coefficients.

Much more work of this sort is essential, given that this scenario looks like an *arm's race* in which Steganographers and Steganalyzers compete to produce better approaches in a technological escalation.

In the Stegi@Work section, we present a common framework that allows us to combine most of the state of the art solutions in a compact and efficient way toward the objective of recovering the hidden content.

Some other flaws related to the classifier-based blind or semi-blind approaches are

- The choice of proper features to train the classifier upon is a key step. There is no systematic rule for feature selection. It is mostly a heuristic, trial and error method [24].
- Some classifiers have several parameters that have to be chosen (type of kernels, learning rate, training conditions) making the process a hard task [24].
- To our knowledge, a standard reference set has yet to emerge in the Steganalysis field to allow fair comparison across different approaches. One step in that direction is the work of [154] which presents two controlled data sets to test hidden message detection approaches and the work of [79] which presents a new benchmark for binary steganalysis methods.

Stegi@Work

What is needed for today's forensics applications is a scalable framework that is able to process a large volume of images (the sheer volume of images on sites such as Flickr and Picasa is testament to this). As we have repeatedly seen throughout this paper, individual techniques for forensic analysis have been developed for specific tools, image characteristics, and imaging hardware, with results presented in the limited capacity of each individual work's focus. If a high capacity framework for digital image forensics was available, the forensic tools presented in this paper could be deployed in a common way, allowing the application of many tools against a candidate image, with the fusion of results giving a high-confidence answer as to whether an image contains steganographic content, is a forgery, or has been produced by a particular imaging system. In our own work in the "Vision of the Unseen," we have focused on the development of a cross-platform distributed framework specifically for Steganalysis, embodying the above ideas, that we call *Stegi@Work*. In this section, we will summarize the overall architecture and capabilities of the Stegi@Work framework as an example of what a distributed forensics framework should encompass.

Stegi@Work, at the highest architectural level (details in Figure 2.16), consists of three entities. A requester client issues jobs for the system to process. Each job consists of a file that does or does not contain steganographic content. This file is transmitted to the Stegi server, which in turn, dispatches the job's processing to the worker clients. Much like other distributed computing frameworks such as *Seti@home*[10] and *Folding@home*[11], worker clients can be ordinary workstations on a network with CPU cycles to spare. The Stegi server collects the results for each job, and performs fusion over the set of results, to come to a final conclusion about the status of the file in question. Each network entity may be connected via a LAN, or logically separated by firewalls in a WAN, facilitating the use of worker clients or requestor clients on a secure or classified network, while maintaining presence on an insecure network, such as the Internet. The Stegi server exists as the common point of contact for both.

The specifics of job communication (details in Figure 2.17), include the specific definitions for each job packet transmitted between network entities. Between the requester client and the Stegi server, both job request and job results packets are exchanged. In a job request, the file in question is transmitted to the server, along with optional tool selection and response requests. If these are not specified, the server can choose them automatically based on the type of the submitted file, as well as a defined site policy. The server receives a detailed report packet from each worker client, including the results of all of the tools applied against a file, as well as additional details about the job, such as execution time. Additional status packets are transmitted between all network entities, including server status to a worker client, notifying it that a job (with the file and appropriate tools) is ready, worker client status to the server, indicating the current state of a job, and server status to a worker client indicating what should be known about a job that is in the system.

The Stegi@Work architecture provides tool support for each worker client in the form of a wrapper API around the tool for each native platform. This API defines process handling, process status, and control signaling, allowing the Stegi server full control over each process on each worker client. The current system as implemented supports wrappers written in C/C++, Java, and Matlab, thus supporting a wide range of tools on multiple platforms. Network communication between each native tool on the worker client and the Stegi@Work system is defined via a set of XML messages. We have created wrappers for the popular analysis tools stegdetect[12] and Digital Invisible Ink Toolkit[13], as well as a custom tool supporting signature-based detection, as well as the statistical χ^2 test.

In order for high portability, allowing for many worker clients, the Stegi@Work framework has been implemented in Java, with tool support, as mentioned above, in a variety of different languages. This is accomplished through the use of Java Native Interface[14] (JNI), with Win32 and Linux calls currently supported. The Stegi@Work server is built on top of JBOSS[15], with an Enterprise Java Beans[16] (EJB) 3.0 object model for all network entities. GUI level dialogues are available for system control at each entity throughout the framework.

The actual use cases for a system like Stegi@Work extend beyond large-scale forensics for

[10] `http://setiathome.berkeley.edu/`

[11] `http://folding.stanford.edu/`

[12] `http://www.outguess.org/detection.php`

[13] `http://diit.sourceforge.net/`

[14] `http://swik.net/JNI+Tutorial`

[15] `http://www.jboss.org/`

[16] `http://www.conceptgo.com/gsejb/index.html`

intelligence or law enforcement purposes. Corporate espionage remains a critical threat to business, with loss estimates as high as $200 billion[17]. An enterprise can deploy requestor clients at the outgoing SMTP servers to scan each message attachment for steganographic content. If such content is detected, the system can quarantine the message, issue alerts, or simply attempt to destroy [74, 132] any detected content automatically, and send the message back on its way. This last option is desirable in cases where false positives are more likely, and thus, a problem for legitimate network users. Likewise, a government agency may choose to deploy the system in the same manner to prevent the theft of very sensitive data.

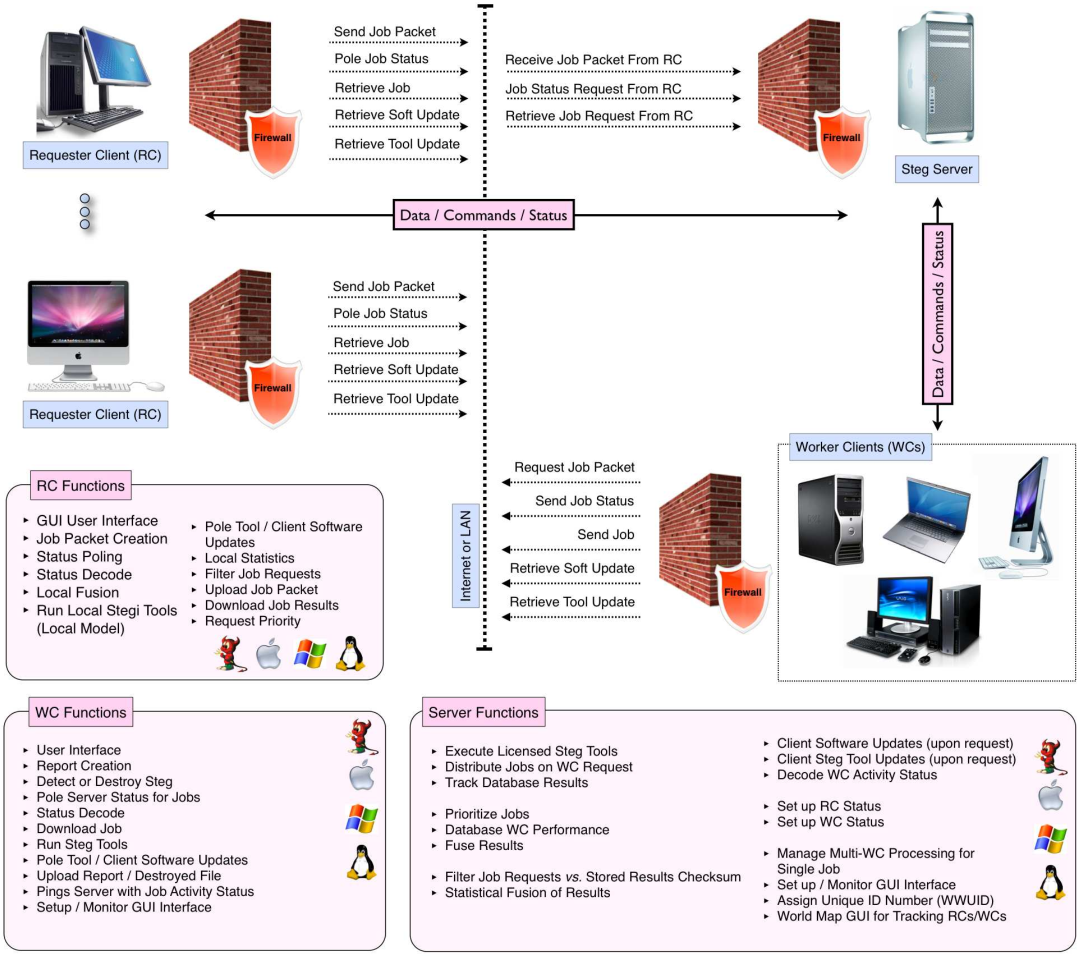


Figure 2.16: Stegi@Work overall architecture.

[17] `http://news.bbc.co.uk/2/hi/technology/5313772.stm`

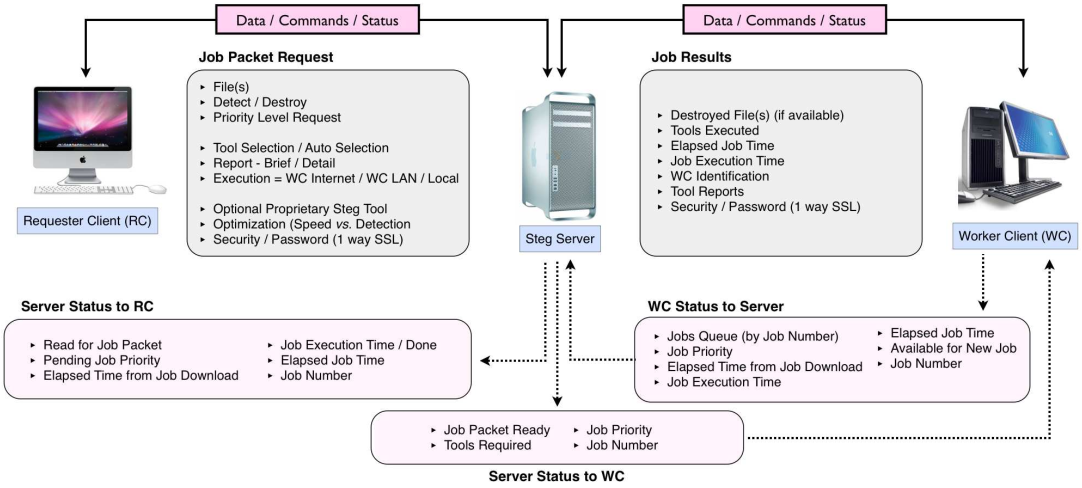


Figure 2.17: Stegi@Work communications architecture.

2.3 Conclusions

A remarkable demand for image-based forensics has emerged in recent years in response to a growing need for investigative tools for a diverse set of needs. From the law enforcement community's perspective, image based analysis is crucial for the investigation of many crimes, most notably child pornography. Yet, crime that utilizes images is not limited to just pornography, with entities as diverse as Colombian drug cartels taking advantage of steganography to mask their activities. From the intelligence community's perspective, the ability to scan large amounts of secret and public data for tampering and hidden content is of interest for strategic national security. As the case of the Iranian missiles has shown, state based actors are just as willing to abuse image processing as common criminals.

But the obvious crimes are not necessarily the most damaging. The digital world presents its denizens with a staggering number of images of dubious authenticity. Disinformation via the media has been prevalent throughout the last century, with doctored images routinely being used for political propaganda. But now, with the near universal accessibility of digital publishing, disinformation has spread to commercial advertising, news media, and the work of malicious pranksters. Is it at all possible to determine whether an image is authentic or not? If we cannot determine the authenticity, what are we to believe about the information the image represents?

Digital Image and Video Forensics research aims at uncovering and analyzing the underlying facts about an image/video. Its main objectives comprise: tampering detection (cloning, healing, retouching, splicing), hidden messages detection/recovery, and source identification with no prior measurement or registration of the image (the availability of the original reference image or video). In this paper, we have taken a look at many individual algorithms and techniques designed for very specific detection goals. However, the specific nature of the entire body of digital image and video forensics work is its main limitation at this point in time. How is an investigator able to choose the correct method for an image at hand? Moreover, the shear magnitude of images that proliferate throughout the Internet poses a serious challenge for large-scale hidden content detection or authenticity verification.

In response to this challenge, we make several recommendations. First, work on decision level and temporal fusion serves as an excellent basis for operational systems. Combining information from many algorithms and techniques yields more accurate results — especially when we do not know precisely what we are looking for. Second, the need for large distributed (or clustered) systems for parallel evaluation fills an important role for national and corporate security. Our Stegi@Work system is an example of this. Third, the evaluation of existing and new algorithms must be improved. The analysis of detection results in nearly all papers surveyed lacks the rigor found in other areas of digital image processing and computer vision, making the assessment of their utility difficult. More troubling, in our paper, only a few papers on counter-forensics for image based forensics were found, leading us to question the robustness of much of the work presented here to a clever manipulator. Finally, for forgery detection and steganalysis, more powerful algorithms are needed to detect specifics about manipulations found in images, not just that an image has been tampered with. Despite these shortcoming, the advancement of the state of the art will continue to improve our *Vision of the Unseen*.

2.4 Acknowledgments

We would like to thank the financial support of Fapesp (Contracts 05/58103-3 and 07/52015-0), CNPq (Contracts 309254/2007-8 and 551007/2007-9), and United States Air Force STTR (Contract FA9550-05-C-0172).

Esteganografia e Esteganálise nos Meios Digitais

No Capítulo 3, discutimos algumas das principais técnicas para o mascaramento digital de informações e para a detecção de mensagens escondidas em imagens.

Mostramos que 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.

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* [64, 113].

Desta forma, é importante desenvolvermos algoritmos para detectar a existência de mensagens escondidas. Neste contexto, aparece a **esteganálise digital**, que se refere ao conjunto de técnicas que são desenvolvidas para distinguir entre objetos que possuem conteúdo escondido (estego-objetos) daqueles que não o possuem (não-estego).

Finalmente, apresentamos as principais tendências relacionadas à Esteganografia e Esteganálise digitais bem como algumas oportunidades de pesquisa.

O trabalho apresentado no Capítulo 3 é o resultado de nosso artigo [149] na *Revista de Informática Teórica e Aplicada* (RITA).

Chapter 3

Steganography and Steganalysis in Digital Multimedia: Hype or Hallelujah?

Abstract

In this paper, we introduce the basic theory behind Steganography and Steganalysis, and present some recent algorithms and developments of these fields. We show how the existing techniques used nowadays are related to Image Processing and Computer Vision, point out several trendy applications of Steganography and Steganalysis, and list a few great research opportunities just waiting to be addressed.

3.1 Introduction

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

More than just a science, *Steganography* is the art of secret communication. Its purpose is to hide the presence of communication, a very different goal than *Cryptography*, that aims to make communication unintelligible for those that do not possess the correct access rights [6]. Applications of Steganography can include feature location (identification of subcomponents within a data set), captioning, time-stamping, and tamper-proofing (demonstration that original contents have not been altered). Unfortunately, not all applications are harmless, and there are strong indications that Steganography has been used to spread child pornography pictures on the internet [64, 113].

In this way, it is important to study and develop algorithms to detect the existence of hidden messages. *Digital Steganalysis* is the body of techniques that attempts to distinguish between

[1] *The effort of secret communication without raising suspicions.*

non-stego or *cover objects*, those that do not contain a hidden message, and *stego-objects*, those that contain a hidden message.

Steganography and Steganalysis have received a lot of attention around the world in the past few years. Some are interested in securing their communications through hiding the very own fact that they are exchanging information. On the other hand, others are interested in detecting the existence of these communications — possibly because they might be related to illegal activities.

In this paper, we introduce the basic theory behind Steganography and Steganalysis, and present some recent algorithms and developments of these fields. We show how the existing techniques used nowadays are related to Image Processing and Computer Vision, point out several trendy applications of Steganography and Steganalysis, and list a few great research opportunities just waiting to be addressed.

The remainder of this paper is organized as follows. In Section 3.2, we introduce the main concepts of Steganography and Steganalysis. Then, we present historical remarks and social impacts in Sections 3.3 and 3.4, respectively. In Section 3.5, we discuss information hiding for scientific and commercial applications. In Sections 3.6 and 3.7, we point out the main techniques of Steganography and Steganalysis. In Section 3.8, we present common-available information hiding tools and software. Finally, in Sections 3.9 and 3.10, we point out open research topics and conclusions.

3.2 Terminology

According to the general model of *Information Hiding*: *embedded data* is the message we want to send secretly. Often, we hide the embedded data in an innocuous medium, called *cover message*. There are many kinds of cover messages such as *cover text*, when we use text to hide a message; or *cover image*, when we use an image to hide a message. The embedding process produces a *stego object* which contains the hidden message. We can use a *stego key* to control the embedding process, so we can also restrict detection and/or recovery of the embedded data to other parties with the appropriate permissions to access this data.

Figure 3.1 shows the process of hiding a message in an image. First we choose the data we want to hide. Further, we use a selected key to hide the message in a previously selected cover image which produces the stego image.

When designing information hiding techniques, we have to consider three competing aspects: capacity, security, and robustness [144]. *Capacity* refers to the amount of information we can embed in a cover object. *Security* relates to an eavesdropper's inability to detect the hidden information. *Robustness* refers to the amount of modification the stego-object can withstand before an adversary can destroy the information [144]. Steganography strives for high security and capacity. Hence, a successful *attack* to the Steganography consists of the detection of the hidden content. On the other hand, in some applications, such as *watermarking*, there is the additional requirement of robustness. In these cases, a successful attack consists in the detection and removal of the copyright marking.

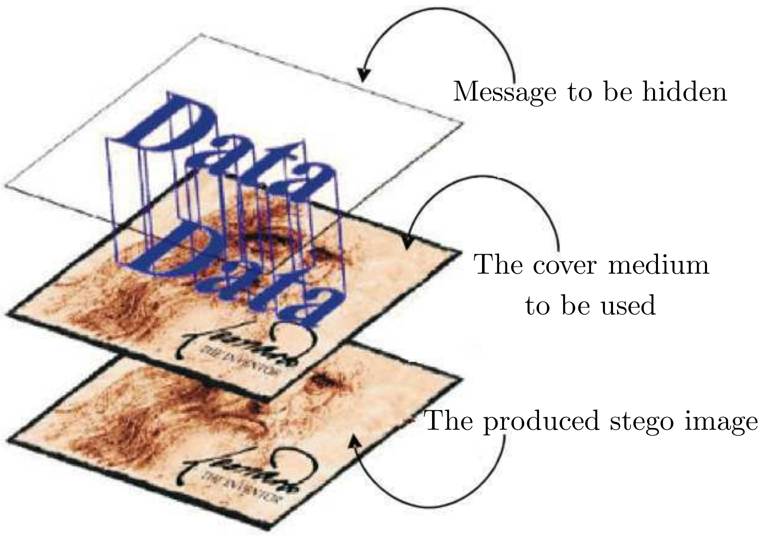


Figure 3.1: A data hiding example.

Figure 3.2 presents the Information Hiding hierarchy [135]. *Covert channels* consist of the use of a secret and secure channel for communication purposes (e.g., military covert channels). *Steganography* is the art, and science, of hiding the information to avoid its detection. It derives from the Greek *steganos* $\sim$ "hide, embed" and *graph* $\sim$ "writing".

We classify Steganography as *technical* and *linguistic*. When we use physical means to conceal the information, such as invisible inks or micro-dots, we are using *technical Steganography*. On the other hand, if we use only "linguistic" properties of the cover object, such as changes in image pixels or letter positions, in a cover text we are using *linguistic Steganography*.

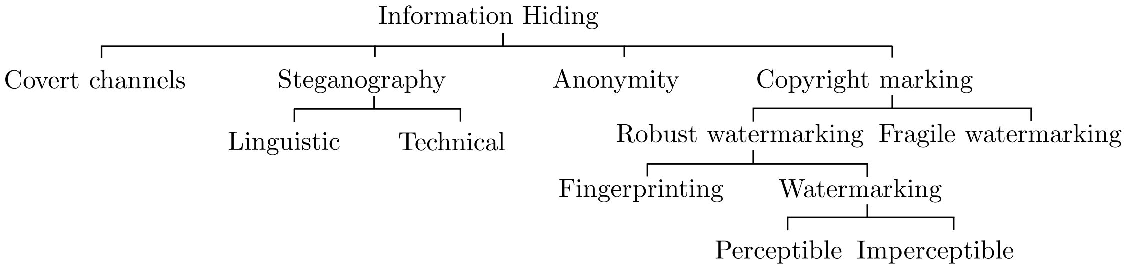


Figure 3.2: Information Hiding hierarchy.

Copyright marking refers to the group of techniques devised to identify the ownership of intellectual property over information. It can be *fragile*, when any modification on the media leads to the loss of the marking; or *robust*, when the marking is robust to some destructive attacks.

Robust copyright marking can be of two types: *fingerprinting* and *watermarking*. *Fingerprinting* hides an unique identifier of the customer who originally acquired the information, recording in the media its ownership. If the copyright owner finds the document in the possession of an unwanted party, she can use the fingerprint information to identify, and prosecute, the customer who violated the license agreement.

Unlike fingerprints, *watermarks* identify the copyright owner of the document, not the identity of the customer. Furthermore, we can classify watermarking according to its visibility to the naked eye as *perceptible* or *imperceptible*.

In short, fingerprints are used to identify violators of the license agreement, while watermarks help with prosecuting those who have an illegal copy of a digital document [131, 135].

Anonymity is the body of techniques devised to surf the *Web* secretly. This is done using sites like *Anonymizer*[2] or *remailers* (blind e-mailing services).

3.3 Historical remarks

Throughout history, people always have aspired to more privacy and security for their communications [77, 122]. One of the first documents describing Steganography comes from *Histories* by Herodotus, the Father of History. In this work, Herodotus gives us several cases of such activities. A man named Harpagus killed a hare and hid a message in its belly. Then, he sent the hare with a messenger who pretended to be a hunter [122].

In order to convince his allies that it was time to begin a revolt against Medes and the Persians, Histaieus shaved the head of his most trusted slave, tattooed the message on his head and waited until his hair grew back. After that, he sent him along with the instruction to shave his head only in the presence of his allies.

Another technique was the use of tablets covered by wax, first used by Demeratus, a Greek who wanted to report from the Persian court back to his friends in Greece that Xerxes, the Great, was about to invade them. The normal use of wax tablets consisted in writing the text in the wax over the wood. Demeratus, however, decided to melt the wax, write the message directly to the wood, and then put a new layer of wax on the wood in such a way that the message was not visible anymore. With this ingenious action, the tablets were sent as apparently blank tablets to Greece. This worked for a while, until a woman named Gorgo guessed that maybe the wax was hiding something. She removed the wax and became the first woman cryptanalyst in History.

During the Renaissance, the Harpagus' hare technique was "improved" by Giovanni Porta, one of the greatest cryptologists of his time, who proposed feeding a message to a dog and then killing the dog [77].

Drawings were also used to conceal information. It is a simple matter to hide information by varying the length of a line, shadings, or other elements of the picture. Nowadays, we have

[2] www.anonymizer.com

proof that great artists, such as Leonardo Da Vinci, Michelangelo, and Rafael, have used their drawings to conceal information [77]. However, we still do not have any means to identify the real contents, or even intention, of these messages.

Sympathetic inks were a widespread technique. Who has not heard about lemon-based ink during childhood? With this type of ink, it is possible to write an innocent letter having a very different message written between its lines.

Science has developed new chemical substances that, combined with other substances, cause a reaction that makes the result visible. One of them is *gallotanic acid*, made from gall nuts, that becomes visible when coming in contact with *copper sulfate* [137].

With the continuous improvement of lenses, photo cameras, and films, people were able to reduce the size of a photo down to the size of a printed period [77, 122]. One such example is micro-dot technology, developed by the Germans during the Second World War, referred to as the "enemy's masterpiece of espionage" by the FBI's director J. Edgar Hoover. Micro-dots are photographs the size of a printed period that have the clarity of standard-sized typewritten pages. Generally, micro-dots were not hidden, nor encrypted messages. They were just so small as to not draw attention to themselves. The micro-dots allowed the transmission of large amounts of data (e.g., texts, drawings, and photographs) during the war.

There are also other forms of hidden communications, like *null ciphers*. Using such techniques, the real message is "camouflaged" in an innocuous message. The messages are very hard to construct and usually look like strange text. This strangeness factor can be reduced if the constructor has enough space and time. A famous case of a null cipher is the book *Hypteronomachia Poliphili* of 1499. A Catholic priest named Colona decided to declare his love to a young lady named Polya by putting the message "Father Colona Passionately loves Polia" in the first letter of each chapter of his book.

3.4 Social impacts

Science and technology changed the way we lived in the 20^{th} century. However, this progress is not without risk. Evolution may have a high social impact, and digital Steganography is no different.

Over the past few years, Steganography has received a lot of attention. Since September 11^{th}, 2001, some researchers have suggested that Osama Bin Laden and Al Qaeda used Steganography techniques to coordinate the World Trade Center attacks. Several years later, nothing was proved [21, 83, 147, 184]. However, since then, Steganography has been a hype.

As a matter of fact, it is important to differentiate what is merely a suspicion from what is real — the hype or the hallelujah. There are many legal uses for Steganography and Steganalysis, as we show in Section 3.5. For instance, we can employ Steganography to create smart data structures and robust watermarking to track and authenticate documents, to communicate privately, to manage digital elections and electronic money, to produce advanced medical imagery, and to devise modern transit radar systems. Unfortunately, there are also illegal uses of these techniques. According to the *High Technology Crimes Annual Report* [108, 120], Steganography and Steganalysis can be used in conjunction with dozens of other cyber-crimes such as: fraud and theft, child pornography, terrorism, hacking, online defamation, intellectual property offenses, and online harassment. There are strong indications that Steganography has been used to spread child pornography pictures on the internet [64, 113].

In this work, we present some possible techniques and legal applications of Steganography and Steganalysis. Of course, the correct use of the information therein is all part of the reader's responsibility.

3.5 Scientific and commercial applications

In this section, we show that there are many applications for Information Hiding.

- **Advanced data structures**. We can devise data structures to conceal unplanned information without breaking compatibility with old software. For instance, if we need extra information about photos, we can put it in the photos themselves. The information will travel with the photos, but it will not disturb old software that does not know of its existence. Furthermore, we can devise advanced data structures that enable us to use small pieces of our hard disks to secretly conceal important information [63, 125].

- **Medical imagery**. Hospitals and clinical doctors can put together patient's exams, imagery, and their information. When a doctor analyzes a radiological exam, the patient's information is embedded in the image, reducing the possibility of wrong diagnosis and/or fraud. Medical-image steganography requires extreme care when embedding additional data within the medical images: the additional information must not affect the image quality [85, 157].

- **Strong watermarks**. Creators of digital content are always devising techniques to describe the restrictions they place on their content. These technique can be as simple as the message "Copyright 2007 by Someone" [188], as complex as the digital rights management system (DRM) devised by Apple Inc. in its iTunes store's contents [175], or the watermarks in the contents of the Vatican Library [110].

- **Military agencies**. Militaries' actions can be based on hidden and protected communications. Even with crypto-graphed content, the detection of a signal in a modern battlefield can lead to the rapid identification and attack of the involved parties in the communication. For this reason, military-grade equipment uses modulation and spread spectrum techniques in its communications [188].

- **Intelligence agencies**. Justice and Intelligence agencies are interested in studying these technologies, and identifying their weaknesses to be able to detect and track hidden messages [64, 109, 113].

- **Document tracking tools**. We can use hidden information to identify the legitimate owner of a document. If the document is leaked, or distributed to unauthorized parties, we can track it back to the rightful owner and perhaps discover which party has broken the license distribution agreement [188].

- **Document authentication**. Hidden information bundled into a document can contain a digital signature that certifies its authenticity [188].

- **General communication**. People are interested in these techniques to provide more security in their daily communications [184, 188]. Many governments continue to see the internet, corporations, and electronic conversations as an opportunity for surveillance [164].

- **Digital elections and electronic money**. Digital elections and electronic money are based on secret and anonymous communications techniques [135, 188].

- **Radar systems**. Modern transit radar systems can integrate information collected in a radar base station, avoiding the need to send separate text and pictures to the receiver's base stations.

- **Remote sensing**. Remote sensing can put together vector maps and digital imagery of a site, further improving the analysis of cultivated areas, including urban and natural sites, among others.

3.6 Steganography

In this section, we present some of the most common techniques used to embed messages in digital images. We choose digital images as cover objects because they are more related to Computer Vision and Image Processing. However, these techniques can be extended to other types of digital media as cover objects, such as text, video, and audio files.

In general, steganographic algorithms rely on the replacement of some noise component of a digital object with a pseudo-random secret message [6]. In digital images, the most common noise component is the least significant bits (LSBs). To the human eye, changes in the value of the LSB are imperceptible, thus making it an ideal place for hiding information without any perceptual change in the cover object.

The original LSB information may have statistical properties, so changing some of them could result in the loss of those properties. Thus, we have to embed the message mimicking the characteristics of the cover bits' [137]. One possibility is to use a *selection method* in which we generate a large number of cover messages in the same way, and we choose the one having the secret embedded in it. However, this method is computationally expensive and only allows small embeddings. Another possibility is to use a *constructive method*. In this approach, we build a mimic function that also simulates characteristics of the cover bits noise.

Generally, both the sender and the receiver share a secret key and use it with a keystream generator. The key-stream is used for selecting the positions where the secret bits will be embedded [137].

Although LSB embedding methods hide data in such a way that humans do not perceive it, these embeddings often can be easily destroyed. As LSB embedding takes place on noise, it is likely to be modified, and destroyed, by further compression, filtering, or a less than perfect format or size conversion. Hence, it is often necessary to employ sophisticated techniques to improve embedding reliability as we describe in Section 3.6.3. Another possibility is to use techniques that take place on the most significant parts of the digital object used. These techniques must be very clever in order to not modify the cover object making the alterations imperceptible.

3.6.1 LSB insertion/modification

Among all message embedding techniques, LSB insertion/modification is a difficult one to detect [6, 147, 188], and it is imperceptible to humans [188]. However, it is easy to destroy [147]. A typical color image has three channels: red, green and blue (R,G,B); each one offers one possible bit per pixel to the hiding process.

In Figure 3.3, we show an example of how we can possibly hide information in the LSB fields. Suppose that we want to embed the bits **1110** in the selected area. In this example, without loss of generality, we have chosen a gray-scale image, so we have one bit available in each image pixel for the hiding process. If we want to hide four bits, we need to select four pixels. To perform the embedding, we tweak the selected LSBs according to the bits we want to hide.

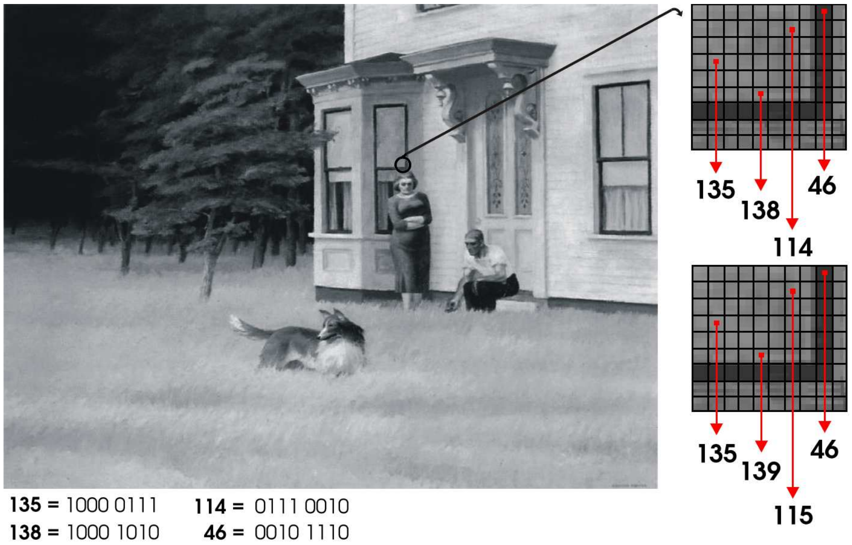


Figure 3.3: The LSB embedding process.

3.6.2 FFTs and DCTs

A very effective way of hiding data in digital images is to use a Direct Cosine Transform (DCT), or a Fast Fourier Transform (FFT), to hide the information in the frequency domain. The DCT algorithm is one of the main components of the JPEG compression technique [60]. In general, DCT and FFT work as follows:

1. Split the image into 8×8 blocks.
2. Transform each block via a DCT/FFT. This outputs a multi-dimensional array of 64 coefficients.
3. Use a quantizer to round each of these coefficients. This is essentially the compression stage and it is where data is lost. Small unimportant coefficients are rounded to 0 while larger ones lose some of their precision.
4. At this stage you should have an array of streamlined coefficients, which are further compressed via a Huffman encoding scheme or something similar.
5. To decompress, use the inverse DCT/FFT.

The hiding process using a DCT/FFT is useful because anyone that looks at pixel values of the image would be unaware that anything is different [188].

Least significant coefficients.

It is possible to use LSB of the quantized DCT/FFT coefficients as redundant bits, and embed the hidden message there. The modification of a single DCT/FFT coefficient affects all 64 image pixels in the block [144]. Two of the simpler frequency-hiding algorithms are JSteg [179] and Outguess [142].

JSteg, Algorithm 2, sequentially replaces the least significant bit of DCT, or FFT, coefficients with the message's data. The algorithm does not use a shared key, hence, anyone who knows the algorithm can recover the message's hidden bits.

On the other hand, Outguess, Algorithm 3, is an improvement over JSteg, because it uses a pseudo-random number generator (PRNG) and a shared key as the PRNG's seed to choose the coefficients to be used.

Block tweaking.

It is possible to hide data during the quantization stage [188]. If we want to encode the bit value 0 in a specific 8×8 square of pixels, we can do this by making sure that all the coefficients are even in such a block, for example by tweaking them. In a similar approach, bit value 1 can be stored by tweaking the coefficients so that they are odd.

With the block tweaking technique, a large image can store some data that is quite difficult to destroy when compared to the LSB method. Although this is a very simple method and works well in keeping down distortions, it is vulnerable to noise [6, 188].

Algorithm 2 JSteg general algorithm

```
Require: message $M$, cover image $I$;
procedure JSteg(M, I)
    while $M \neq$ NULL do
        get next DCT coefficient from $I$;
        if DCT $\neq 0$ and DCT $\neq 1$ then          ▷ We only change non-0/1 coefficients
            b $\leftarrow$ next bit from $M$;
            replace DCT LSB with message bit $b$;
            $M \leftarrow M - b$;
        end if
        Insert DCT into stego image $S$;
    end while
    return $S$;
end procedure
```

Algorithm 3 Outguess general algorithm

```
Require: message $M$, cover image $I$, shared key $k$;
procedure Outguess(M, I, k)
    Initialize PRNG with the shared key $k$
    while $M \neq$ NULL do
        get pseudo-random DCT coefficient from $I$;
        if DCT $\neq 0$ and DCT $\neq 1$ then          ▷ We only change non-0/1 coefficients
            b $\leftarrow$ next bit from $M$;
            replace DCT LSB with message bit $b$;
            $M \leftarrow M - b$;
        end if
        Insert DCT into stego image $S$;
    end while
    return $S$;
end procedure
```

Coefficient selection.

This technique consists of the selection of the k largest DCT or FFT coefficients $\{\gamma_1 \ldots \gamma_k\}$ and modify them according to a function f that also takes into account a measure α of the required strength of the embedding process. Larger values of α are more resistant to error, but they also introduce more distortions.

The selection of the coefficients can be based on visual significance (e.g., given by zigzag ordering [188]). The factors α and k are user-dependent. The function $f(\cdot)$ can be

$$f(\gamma_i') = \gamma_i + \alpha b_i, \tag{3.1}$$

where b_i is a bit we want to embed in the coefficient γ_i.

Wavelets.

DCT/FFT transformations are not so effective at higher-compression levels. In such scenarios, we can use wavelet transformations instead of DCT/FFTs to improve robustness and reliability.

Wavelet-based techniques work by taking many wavelets to encode a whole image. They allow images to be compressed by storing the high and low frequency details separately in the image. We can use the low frequencies to compress the data, and use a quantization step to compress even more. Information hiding techniques using wavelets are similar to the ones with DCT/FFT [188].

3.6.3 How to improve security

Robust Steganography systems must observe the Kerckhoffs' Principle [160] in Cryptography, which holds that a cryptographic system's security should rely solely on the key material. Furthermore, to remain undetected, the unmodified cover medium used in the hiding process must be kept secret or destroyed. If it is exposed, a comparison between the cover and stego media immediately reveals the changes.

Further procedures to improve security in the hiding process are:

- **Cryptography**. Steganography supplements Cryptography, it does not replace it. If a hidden message is encrypted, it must also be decrypted if discovered, which provides another layer of protection [73].

- **Statistical profiling**. Data embedding alters statistical properties of the cover medium. To overcome such alterations, the embedding procedure can learn the statistics about the cover medium in order to minimize the amount of changes. For instance, for each bit changed to zero, the embedding procedure changes another bit to one.

- **Structural profiling**. Mimicking the statistics of a file is just the beginning. We can use the structure of the cover medium to better hide the information. For instance, if our cover medium is an image of a person, we can choose regions of this image that are rich in details such as the eyes, mouth and nose. These areas are more resilient to compression and conversion artifacts [60].

- **Change of the order**. Change the order in which the message is presented. The order itself can carry the message. For instance, if the message is a list of items, the order of the items can itself carry another message.

- **Split the information**. We can split the data into any number of packets and send them through different routes to their destination. We can apply sophisticated techniques in order to need only k out of n parts to reconstruct the whole message [188].

- **Compaction**. Less information to embed means fewer changes in the cover medium, lowering the probability of detection. We can use compaction to shrink the message and the amount of needed alterations in the cover medium.

3.7 Steganalysis

With the indications that steganography techniques have been used to spread child pornography pictures on the internet [64, 113], there is a need to design and evaluate powerful detection techniques able to avoid or minimize such actions. In this section, we present an overview of current approaches, attacks, and statistical techniques available in Steganalysis.

Steganalysis refers to the body of techniques devised to detect hidden contents in digital media. It is an allusion to Cryptanalysis which refers to the body of techniques devised to break codes and cyphers [160].

In general, it is enough to detect whether a message is hidden in a digital content. For instance, law enforcement agencies can track access logs of hidden contents to create a network graph of suspects. Later, using other techniques, such as physical inspection of apprehended material, they can uncover the actual contents and apprehend the guilty parties [73,147]. There are three types of Steganalysis attacks: (1) aural; (2) structural; and (3) statistical.

1. **Aural attacks**. They consist of striping away the significant parts of a digital content in order to facilitate a human's visual inspection for anomalies [188]. A common test is to show the LSBs of an image.

2. **Structural attacks**. Sometimes, the format of the digital file changes as hidden information is embedded. Often, these changes lead to an easily detectable pattern in the structure of the file format. For instance, it is not advisable to hide messages in images stored in GIF format. In such a format an image's visual structure exists to some degree in all of an image's bit layers due to the color indexing that represents 2^{24} colors using only 256 values [191].

3. **Statistical attacks**. Digital pictures of natural scenes have distinct statistical behavior. With proper statistical analysis, we can determine whether or not an image has been

altered, making forgeries mathematically detectable [109]. In this case, the general purpose of Steganalysis is to collect sufficient statistical evidence about the presence of hidden messages in images, and use them to classify [16] whether or not a given image contains a hidden content. In the following section, we present some available statistical-based techniques for hidden message detection.

3.7.1 χ^2 analysis

Westfeld and Pfitzmann [191] have present χ^2 analysis to detect hidden messages. They showed that an L-bit color channel can represent 2^L possible values. If we split these values into 2^{L-1} pairs which only differ in the LSBs, we are considering all possible patterns of neighboring bits for the LSBs. Each of these pairs is called a *pair of value* (PoV) in the sequence [191].

When we use all the available LSB fields to hide a message in an image, the distribution of odd and even values of a PoV will be the same as the 0/1 distribution of the message bits. The idea of the χ^2 analysis is to compare the theoretically expected frequency distribution of the PoVs with the real observed ones [191]. However, we do not have the original image and thus the expected frequency. In the original image, the theoretically expected frequency is the arithmetical mean of the two frequencies in a PoV. As we know, the embedding function only affects the LSBs, so it does not affect the PoV's distribution after an embedding. Given that, the arithmetical mean remains the same in each PoV, and we can derive the expected frequency through the arithmetic mean between the two frequencies in each PoV.

Westfeld and Pfitzmann [191] have showed that we can apply the χ^2 (chi squared-test) over these PoVs to detect hidden messages. The χ^2 test general formula is

$$\chi^2 = \sum_{i=1}^{\nu+1} \frac{(f_i^{obs} - f_i^{exp})^2}{f_i^{exp}}, \tag{3.2}$$

where ν is the number of analyzed PoVs, f_i^{obs} and f_i^{exp} are the observed frequencies and the expected frequencies respectively.

The probability of hiding, ph, in a region is given by the complement of the cumulative distribution

$$ph = 1 - \int_0^{\chi^2} \frac{t^{(\nu-2)/2} e^{-t/2}}{2^{\nu/2}\Gamma(\nu/2)} dt, \tag{3.3}$$

where $\Gamma(\cdot)$ is the Euler-Gamma function. We can calculate this probability in different regions of the image.

This approach can only detect sequential messages hidden in the first available pixels' LSBs, as it only considers the descriptors' value. It does not take into account that, for different images, the threshold value for detection may be quite distinct [147].

Simply measuring the descriptors constitutes a low-order statistic measurement. This approach can be defeated by techniques that maintain basic statistical profiles in the hiding process [143, 147].

Improved techniques such as Progressive Randomization (PR) [147] addresses the low-order statistics problem by looking at the descriptors' behavior along selected regions (feature regions).

3.7.2 RS analysis

Fridrich et al. have presented RS analysis [50]. It consists of the analysis of the LSB loss-less embedding capacity in color and gray-scale images. The loss-less capacity reflects the fact that the LSB plane — even though it looks random — is related to the other bit planes [50]. Modifications in the LSB plane can lead to statistically detectable artifacts in the other bit planes of the image.

To measure this behavior, Fridrich and colleagues have proposed simulation of artificial new embeddings in the analyzed images using some defined functions.

Let I be the image to be analyzed with width W and height H pixels. Each pixel has values in P. For an 8 bits per pixel image, we have $P = \{0 \ldots 255\}$. We divide I into G disjoint groups of n adjacent pixels. For instance, we can choose $n = 4$ adjacent pixels. We define a discriminant function f responsible to give a real number $f(x_1, \ldots, x_n) \in \Re$ for each group of pixels $G = (x_1, \ldots, x_n)$. Our objective using f is to capture the smoothness of G. Let the discrimination function be

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|. \tag{3.4}$$

Furthermore, let F_1 be a flipping invertible function $F_1 : 0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255$, and F_{-1} be a shifting function $F_{-1} : -1 \leftrightarrow 0, 1 \leftrightarrow 2, \ldots, 255 \leftrightarrow 256$ over P. For completeness, let F_0 be the identity function such as $F_0(x) = x \; \forall \; x \; \in \; P$.

Define a mask $\mathcal{M}$ that represents which function to apply to each element of a group G. The mask $\mathcal{M}$ is an n-tuple with values in $\{-1, 0, 1\}$. The value -1 stands for the application of the function F_{-1}; 1 stands for the function F_1; and 0 stands for the identity function F_0. Similarly, we define $-\mathcal{M}$ as $\mathcal{M}$'s complement.

We apply the discriminant function f with the functions $F_{\{-1,0,1\}}$ defined through a mask $\mathcal{M}$ over all G groups to classify them into three categories:

- **Regular**. $G \in R_{\mathcal{M}} \Leftrightarrow f(F_{\mathcal{M}}(G)) > f(G)$
- **Singular**. $G \in S_{\mathcal{M}} \Leftrightarrow f(F_{\mathcal{M}}(G)) < f(G)$
- **Unusable**. $G \in U_{\mathcal{M}} \Leftrightarrow f(F_{\mathcal{M}}(G)) = f(G)$

Similarly, we classify the groups $R_{-\mathcal{M}}$, $S_{-\mathcal{M}}$, and $U_{-\mathcal{M}}$ for the mask $-\mathcal{M}$. As a matter of fact, it holds that

$$\frac{R_{\mathcal{M}} + S_{\mathcal{M}}}{T} \leq 1 \quad \text{and} \quad \frac{R_{-\mathcal{M}} + S_{-\mathcal{M}}}{T} \leq 1,$$

where T is the total number of G groups.

The method's statistical hypothesis is that, for typical images

$$R_{\mathcal{M}} \approx R_{-\mathcal{M}} \quad \text{and} \quad S_{\mathcal{M}} \approx S_{-\mathcal{M}}.$$

What is interesting is that, in an image with a hidden content, the greater the message size, the greater the $R_{-\mathcal{M}}$ and $S_{-\mathcal{M}}$ difference, and the lower the difference between $R_{\mathcal{M}}$ and $S_{\mathcal{M}}$. This behavior points out to high-probability chance of embedding in the analyzed image [50].

3.7.3 Gradient-energy flipping rate

Li Shi et al. have presented the Gradient-Energy Flipping Rate (GEFR) technique for Steganalysis. It consists in the analysis of the gradient-energy variation due to the hiding process [166].

Let $I(n)$ be an unidimensional signal. The gradient $r(n)$, before the hiding is

$$r(n) = I(n) - I(n-1), \tag{3.5}$$

and the $I(n)$'s gradient energy (GE), is

$$GE = \sum |I(n) - I(n-1)|^2 = \sum r(n)^2. \tag{3.6}$$

After the hiding of a signal $S(n)$ in the original signal, $I(n)$ becomes $I'(n)$ and the gradient becomes

$$\begin{aligned} r(n) &= I(n) - I(n-1) \\ &= (I(n) + S(n)) - (I(n-1) + S(n-1)) \\ &= r(n) + S(n) - S(n-1). \end{aligned} \tag{3.7}$$

The probability distribution function of $S(n)$ is

$$\begin{cases} \rho(S(n)) \approx 0 & = \frac{1}{2} \\ \rho(S(n)) \approx \pm 1 & = \frac{1}{4} \end{cases} \tag{3.8}$$

After any kind of embedding, the new gradient energy GE' is

$$\begin{aligned} GE' &= \sum |r(n)|^2 = \sum |r(n) + S(n) - S(n-1)|^2 \\ &= \sum |r(n) + \Delta(n)|^2, \text{where } \Delta(n) = S(n) - S(n-1). \end{aligned} \tag{3.9}$$

To perform the detection, it is necessary to define a process of inverting the bits of an image's LSB plane. For that, we can use a function F which is similar to the one we described in Section 3.7.2.

Let I be the cover image with $W \times H$ pixels and $p \leq W \times H$ be the size of the hidden message. The application of the function F results in the properties:

- For $p = W \times H$, there is $\dfrac{W \times H}{2}$ pixels with inverted LSB. That means that the embedding rate is 50% and the gradient energy is given by $GE = \left(\dfrac{W \times H}{2}\right)$.

- The original image's gradient energy is given by $EG(0)$. After inverting all available LSBs using F, the gradient energy becomes $GE' = W \times H$.
- For $p < W \times H$, there is $\frac{p}{2}$ pixels with inverted LSB. Let $I(\frac{p}{2})$ be the modified image. The resulting gradient energy is $GE = \frac{p/2}{W \times H} = EG(0) + p$. If F is applied over $I(\frac{p}{2})$, the resulting gradient energy is $EG = \frac{W \times H - p/2}{W \times H}$.

With these properties, Li Shi et al. have proposed the following detection procedure:

1. Find the test image's gradient energy $GE\left(\frac{p/2}{W \times H}\right)$;
2. Apply F over the test image and calculate $GE\left(\frac{W \times H - p/2}{W \times H}\right)$;
3. Find $GE\left(\frac{W \times H}{2}\right) = \left[EG\left(\frac{p/2}{W \times H}\right) + GE\left(\frac{W \times H - p/2}{W \times H}\right)\right]/2$;
4. $GE(0)$ is based on $GE\left(\frac{W \times H}{2}\right) = GE(0) + W \times H$;
5. Finally, the estimated size of the hidden message is given by

$$p' = GE\left(\frac{p/2}{W \times H}\right) - GE(0).$$

3.7.4 High-order statistical analysis

Lyu and Farid [41, 42, 95, 96] have introduced a detection approach based on high-order statistical descriptors. Natural images have regularities that can be detected by high-order statistics through wavelet decompositions [96]. To decompose the images, Lyu and colleagues have used quadrature mirror filters (QMFs) [180]. This decomposition divides the image into multiple scales and orientations resulting in four subbands: vertical, horizontal, diagonal, and low-pass which can be recursively used to produce subsequent scales.

Let $V_i(x, y)$, $H_i(x, y)$, and $D_i(x, y)$ be the vertical, horizontal, and diagonal subbands for a given scale $i \in \{1 \ldots n\}$. Figure 3.4 depicts this process.

From the QMF decomposition, the authors create a statistical model composed of mean, variance, skewness, and kurtosis for all subbands and scales. These statistics characterize the basic coefficients' distribution. The second set of statistics is based on the errors in an optimal linear predictor of coefficient magnitude. The subband coefficients are correlated to their spatial, orientation, and scale neighbors [20]. For illustration purposes, consider first a vertical band, $V_i(x, y)$, at scale i. A linear predictor for the magnitude of these coefficients in a subset of all possible neighbors is given by

$$\begin{aligned} V_i(x,y) &= w_1 V_i(x-1,y) + w_2 V_i(x+1,y) + w_3 V_i(x,y-1) + w_4 V_i(x,y+1) + \\ &\quad + w_5 V_{i+1}(\frac{x}{2},\frac{y}{2}) + w_6 D_i(x,y) + w_7 D_{i+1}(\frac{x}{2},\frac{y}{2}), \end{aligned} \tag{3.10}$$

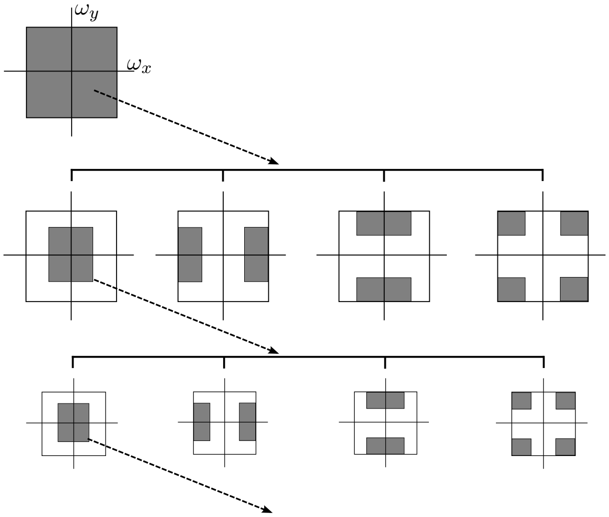


Figure 3.4: QMF decomposition scheme.

where w_k denotes the scalar weighting values. The error coefficients are calculated using quadratic minimization of the error function

$$E(w) = [V - Qw]^2, \tag{3.11}$$

where $w = (w_1, \ldots, w_7)^T$, V is a column vector of magnitude coefficients, and Q is the magnitude neighbors' coefficients as proposed in Equation 3.10. The error function is minimized through differentiation with respect to w

$$\frac{dE(w)}{dw} = 2Q^T[V - Qw]. \tag{3.12}$$

After simplifications, we calculate w_k directly with the linear predictor log error

$$E = \log_2(V) - \log_2(|Qw|). \tag{3.13}$$

With a recursive application of this process to all subbands, scales, and orientation, we have a total of $12(n-1)$ error statistics plus $12(n-1)$ basic ones. This amounts to a $24(n-1)$-sized feature vector. This feature vector feeds a classifier, which is able to output whether or not an unknown image contains a hidden message. Lyu and colleagues have used Linear Discriminant Analysis and Support Vector Machines to perform the classification stage [16].

3.7.5 Image quality metrics

Avcibas et al. have presented a detection scheme based on image quality metrics (IQMs) [1,9,10]. Image quality metrics are often used for coding artifact evaluation, performance prediction of vision algorithms, quality loss due to sensor inadequacy, etc.

Steganographic schemes, whether by spread-spectrum, quantization modulation, or LSB insertion/modification, can be represented as a signal addition to the cover image. In this context, Avcibas and colleagues' hypothesis is that steganographic schemes leave statistical evidences that can be exploited for detection with the aid of IQMs and multivariate regression analysis (ANOVA).

Using ANOVA, the authors have pointed out that the following IQMs are the best feature generators: mean absolute error, mean square error, Czekznowski correlation, image fidelity, cross correlation, spectral magnitude distance, normalized mean square, HVS error, angle mean, median block spectral phase distance, and median block weighted spectral distance.

After measuring the IQMs in a training set of images with and without hidden messages, the authors propose a multivariate normalized regression to values -1 and 1. In the regression model, each decision is expressed by y_i in a set of n observation images and q available IQMs. A linear function of the IQMs is given by

$$\begin{cases} y_1 & = & \beta_1 x_{11} + \beta_2 x_{12} + \ldots + \beta_q x_{1q} + \epsilon_1 \\ y_2 & = & \beta_2 x_{21} + \beta_2 x_{22} + \ldots + \beta_q x_{2q} + \epsilon_2 \\ & \vdots & \\ y_N & = & \beta_n x_{n1} + \beta_2 x_{12} + \ldots + \beta_q x_{nq} + \epsilon_n, \end{cases} \tag{3.14}$$

where x_{ij} is the quality coefficient for the image $i \in \{1 \ldots n\}$ and IQM $j \in \{1 \ldots q\}$. Finally, β_k is the regression coefficient, and ϵ is random error.

Once we calculate these coefficients, we can use the resulting coefficient vector to any new image in order to classify it as stego or non-stego image.

3.7.6 Progressive Randomization (PR)

Rocha and Goldenstein [147] have presented the Progressive Randomization descriptor for Steganalysis. It is a new image descriptor that captures the difference between image classes (with and without hidden messages) using the statistical artifacts inserted during a perturbation process that increases randomness with each step.

Algorithm 4 summarizes the four stages of PR applied to Steganalysis: the randomization process (c.f., Sec. 3.7.6); the selection of feature regions (c.f., Sec. 3.7.6); the statistical descriptors analysis (c.f., Sec. 3.7.6), and invariance (c.f., Sec. 3.7.6).

Pixel perturbation.

Let $\mathbf{x}$ be a Bernoulli distributed random variable with $Prob\{\mathbf{x} = 0\}) = Prob(\{\mathbf{x} = 1\}) = 1/2$, B be a sequence of bits composed by independent trials of $\mathbf{x}$, p be a percentage, and S be a random set of pixels of an input image.

Given an input image I of $|I|$ pixels, we define the LSB pixel perturbation $T(I, p)$ the process of substitution of the LSBs of S of size $p \times |I|$ according to the bit sequence B. Consider a pixel $px_i \in S$ and an associated bit $b_i \in B$

$$\mathcal{L}(px_i) \leftarrow b_i \text{ for all } px_i \in S. \tag{3.15}$$

Algorithm 4 The PR descriptor

```
Require: Input image $I$; Percentages $P = \{P_1, \dots P_n\}$;
Randomization: perform $n$ LSB pixel disturbances of the original image    ▷ Sec. 3.7.6
    $\{O_i\}_{i=0 \dots n.} = \{I, T(I, P_1), \dots, T(I, P_n)\}.$
Region selection: select $r$ feature regions of each image $i \in \{O_i\}_{i=0 \dots n}$    ▷ Sec. 3.7.6
    $\{O_{ij}\}_{\substack{i=0 \dots n, \\ j=1 \dots r.}} = \{O_{01}, \dots, O_{nr}\}.$
Statistical descriptors: calculate $m$ descriptors for each region    ▷ Sec. 3.7.6
    $\{d_{ijk}\} = \{d_k(O_{ij})\}_{\substack{i=0 \dots n, \\ j=1 \dots r, \\ k=1 \dots m.}}$
Invariance: normalize the descriptors based on $I$    ▷ Sec. 3.7.6
    $F = \{f_e\}_{e=1 \dots n \times r \times m} = \left\{ \frac{d_{ijk}}{d_{0jk}} \right\}_{\substack{i=0 \dots n, \\ j=1 \dots r, \\ k=1 \dots m.}}$
Classification. Use $F \in \Re^{n \times r \times m}$ in your favorite machine learning black box.
```

where $\mathcal{L}(px_i)$ is the LSB of the pixel px_i.

The randomization process.

Given an original image I as input, the randomization process consists of the progressive application $I, T(I, P_1), \dots, T(I, P_n)$ of LSB pixel disturbances. The process returns n images that only differ in the LSB from the original image and are identical to the naked eye.

The $T(I, P_i)$ transformations are perturbations of different percentages of the available LSBs. Here, we use $n = 6$ where $P = \{1\%, 5\%, 10\%, 25\%, 50\%, 75\%\}$, $P_i \in P$ denotes the relative sizes of the set of selected pixels S. The greater the LSB pixel disturbance, the greater the resulting LSB entropy of the transformation.

Feature region selection.

Local image properties do not show up under a global analysis [188]. The authors use statistical descriptors of local regions to capture the changing dynamics of the statistical artifacts inserted during the randomization process (c.f., Sec. 3.7.6).

Given an image I, they use r regions with size $l \times l$ pixels to produce localized statistical descriptors (Figure 3.5).

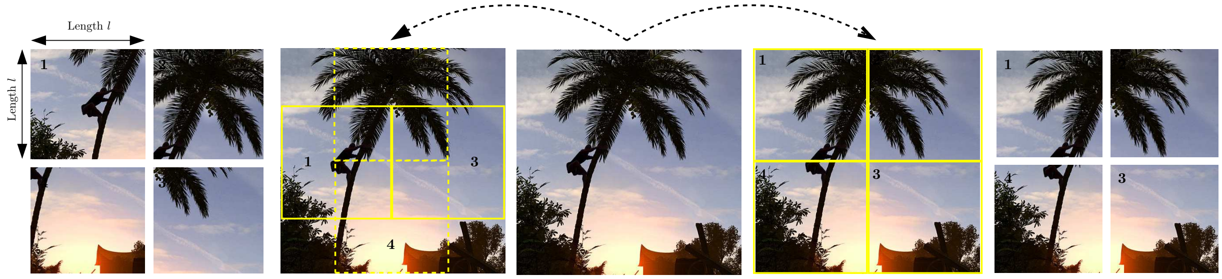


Figure 3.5: The PR eight overlapping regions.

Statistical descriptors.

When we disturb all the available LSBs in S with a sequence B, the distribution of 0/1 values of a PoV (see Section 3.7.1) will be the same as in B. The authors apply the χ^2 (chi-squared test) [191] and U_T (Ueli Maurer Universal Test) [102] to analyze the images.

- χ^2 **test**. The χ^2 test [48] compares two histograms f^{obs} and f^{exp}. Histogram f^{obs} represents the observations and f^{exp} represents the expected histogram. The procedure computes the sum of the square differences of f^{obs} and f^{exp} divided by f^{exp},

$$\chi^2 = \sum_i \frac{(f_i^{obs} - f_i^{exp})^2}{f_i^{exp}}. \tag{3.16}$$

- **Ueli test**. The Ueli test (U_T) [102] is an effective way to evaluate the randomness of a given sequence of numbers. U_T splits an input data S into n blocks. For each block b_i, it analyzes each of the $n-1$ remaining blocks, looks for the most recent occurrence of b_i, and takes the log of the summed temporal occurrences. Let $B(S) = (b_1, b_2, \ldots, b_N)$ be a set of n blocks such that $\cup_{\forall b_i} = S$. Let $|b_i| = L$ be the block size for each i and $|B(S)| = N$ be the number of blocks. We define $U_T : B(S) \rightarrow \Re^+$ as

$$U_T(B(S)) = \frac{1}{K} \sum_{i=Q}^{Q+K} \ln A(b_i), \tag{3.17}$$

where K is the number of analyzed bits (e.g., $K = N$), Q is a shift in $B(S)$ (e.g., $Q = \frac{K}{10}$ [102]), and

$$A(b_i) = \begin{cases} i & \nexists i' \in \mathbb{N}, i' < i | b_{i'} = b_i, \\ \min\{i' : b_{i'} = b_i\} & \text{otherwise.} \end{cases} \tag{3.18}$$

Invariance transformation.

The variation rate of the statistical descriptors is more interesting than their values. The authors propose the normalization of all descriptors from the transformations with regard to their values in the original image I

$$F = \{f_e\}_{e=1\ldots n\times r\times m} \quad = \quad \left\{ \frac{d_{ijk}}{d_{0jk}} \right\}_{\substack{i=0\ldots n,\\ j=1\ldots r,\\ k=1\ldots m.}} \quad , \tag{3.19}$$

where d denotes a descriptor $1 \leq k \leq m$ of a region $1 \leq j \leq r$ of an image $0 \leq i \leq n$, and F is the final generated descriptor vector of the image I.

Classification.

The authors use a labeled set of images to learn the behavior of the selected statistical descriptors and train different classifiers (supervised learning). The goal is to determine whether a new incoming image contains a hidden message. They have trained and validated the technique using a series of classifiers such as CTREES, SVMS, LDA and Bagging ensembles [147].

The statistical hypothesis is that the greater the embedded message, the lower the ratio between subsequent iterations of the progressive randomization operation. Images with no hidden content have different behavior under PR than images that have suffered some process of message embedding [147].

3.8 Freely available tools and software

Many Steganography and Steganalysis applications are freely available on the internet for a great variety of platforms which includes DOS, Windows, Mac OS, Unix, and Linux.

Romana Machado has introduced *Ezstego* and *Stego Online*[3], two tools designed in Java language suitable to Steganography in 8-bits indexed images stored in the GIF format [174].

Henry Hastur has presented two other tools: *Mandelsteg* e *Stealth*[4]. *Mandelsteg* generates fractal images to hide the messages. *Stealth* is a software that uses PGP Cryptography [197] in the embedding process. Two other software tools that incorporate Cryptography in the hiding process are *White Noise Storm*[5] by Ray Arachelian and *S-Tools*[6].

Colin Maroney has devised *Hide and Seek*[7]. This tool is able to hide a list of files in one image. However, it does not use Cryptography. Derek Upham has presented *Jsteg*[8], which is able to hide messages using the DCT/FFT transformed space. Niels Provos has introduced *Outguess*[9] which is an improvement over JSteg-based techniques.

Finally, Anderson Rocha and colleagues have introduced *Camaleão*[10] [151, 152], which uses cyclic permutations and block cyphering to hide messages in the least significant bits of loss-less compression images.

[3] `http://www.stego.com`

[4] `ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/`

[5] `ftp.csua.berkeley.edu/pub/cypherpunks/steganography/wns210.zip`

[6] `ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/s-tools4.zip`

[7] `ftp://csua.berkeley.edu/pub/cypherpunks/steganography/hdsk41b.zip`

[8] `ftp.funet.fi/pub/crypt/steganography`

[9] `http://www.outguess.org/`

[10] `http://andersonrocha.cjb.net`

3.9 Open research topics

When performing data-hiding in digital images, we have an additional problem: images are expected to be subjected to many operations, ranging from simple transformations, such as translations, to nonlinear transformations, such as blurring, filtering, lossy compression, printing, and rescanning. The hidden messages should survive all attacks that do not degrade the image's perceived quality [6].

Steganography's main problem involves designing robust information-hiding techniques. It is crucial to derive approaches that are robust to geometrical attacks as well as nonlinear transformations, and to find detail-rich regions in the image that do not lead to artifacts in the hiding process. The hidden messages should not degrade the perceived quality of the work, implying the need for good image-quality metrics.

Hiding techniques often rely on private key sharing, which involves previous communication. It is important to work on algorithms that use asymmetric key schemes.

If multiple messages are inserted in a single object, they should not interfere with each other [6].

We need new powerful Steganalysis techniques that can detect messages without prior knowledge of the hiding algorithm (blind detection). The detection of very small messages is also a significant problem. Finally, we need adaptive techniques that do not involve complex training stages.

3.10 Conclusions

In this paper, we have presented an overview of the past few years of Steganography and Steganalysis, we have showed some of the most interesting hiding and detection techniques, and we have discussed a series of applications on both topics.

Terrorism has infiltrated the public's perception of this technology for a long period. Public fear created by mainstream press reports, which often featured US intelligence agents claiming that terrorists were using Steganography, created a mystique around data hiding techniques. Legislators in several US states have either considered or passed laws prohibiting the use and dissemination of technology to conceal data [61].

Six years after September 11^{th}, 2001's tragic incidents, Steganography and Steganalysis have become mature disciplines, and data hiding approaches have outlived their period of hype. Public perception should now move beyond the initial notion that these techniques are suitable only for terrorist-cells' communications. Steganography and Steganalysis have many legitimate applications, and represent great research opportunities waiting to be addressed.

3.11 Acknowledgments

We thank the support of FAPESP (05/58103-3 and 07/52015-0) and CNPq (301278/2004 and 551007/2007-9). We also thank Dr. Valerie Miller for proof reading this article.

Esteganálise e Categorização de Imagens

No Capítulo 4, apresentamos uma abordagem para meta-descrição de imagens denominada Randomização Progressiva (PR) para nos auxiliar nos problemas de: (1) Detecção de mensagens escondidas em imagens digitais; e (2) Categorização de imagens.

PR é um novo meta-descritor que captura as diferenças entre classes gerais de imagens usando os artefatos estatísticos inseridos durante um processo de perturbação sucessiva das imagens analisadas.

Nossos experimentos mostram que esta técnica captura bem a separabilidade de algumas classes de imagens. A observação mais importante é que classes diferentes de imagens possuem comportamentos distintos quando submetidas a sucessivas perturbações. Por exemplo, um conjunto de imagens que não possui mensagens escondidas apresenta diferentes artefatos mediante sucessivas perturbações comparado com um conjunto de imagens que possui mensagens escondidas.

Testamos a técnica no contexto da análise forense de imagens para detecção de mensagens escondidas bem como para a classificação geral de imagens em categorias como *indoors*, *outdoors*, *geradas em computador* e *obras de arte*.

O trabalho apresentado no Capítulo 4 é uma compilação de nosso artigo submetido à *Elsevier Computer Vision and Image Understanding* (CVIU). Após um estudo que mostrou viabilidade comercial de nossa técnica, conseguimos o depósito de uma patente nacional junto ao INPI e sua versão internacional junto ao PCT.

Finalmente, o trabalho de detecção de mensagens nos rendeu a publicação [147] no *IEEE Intl. Workshop on Multimedia and Signal Processing (MMSP)*. A extensão da técnica para o cenário multi-classe (*indoors*, *outdoors*, *geradas em computador*, e *obras de arte*) resultou o artigo [148] no *IEEE Intl. Conference on Computer Vision (ICCV)*.

Chapter 4

Progressive Randomization: Seeing the Unseen

Abstract

In this paper, we introduce the Progressive Randomization (PR): a new image meta-description approach suitable for different image inference applications such as broad class *Image Categorization* and *Steganalysis*. The main difference among PR and the state-of-the-art algorithms is that it is based on progressive perturbations on pixel values of images. With such perturbations, PR captures the image class separability allowing us to successfully infer high-level information about images. Even when only a limited number of training examples are available, the method still achieves good separability, and its accuracy increases with the size of the training set. We validate our method using two different inference scenarios and four image databases.

4.1 Introduction

In many real-life applications, we have to make decisions based only on images. In this paper, we introduce a new image meta-description approach based only on information invisible to the naked eye. We apply and validate our new technique on two very distinct problems that use supervised learning: *Image Categorization* and *Digital Steganalysis*.

Image Categorization is the body of techniques that distinguish between image classes, pointing out the global semantic type of an image. Here, we want to distinguish the class of an image (e.g., *Indoors* from *Outdoors*), or the type of an object in restricted domains (e.g., vegetables in a supermarket cashier). One possible scenario for a consumer application is to group a photo album, automatically, according to classes. Another situation can be to automate a supermarket cashier. Common techniques in content-based image retrieval use color histograms and texture [60], bag of features [62, 68, 100], and shape and layout measures [196] to perform queries in massive image databases. With our solution, we can improve these techniques by automatically restraining the search to one or more classes.

Digital Steganalysis is the body of techniques that attempts to distinguish between *non-stego* or *cover objects*, those that do not contain a hidden message, and *stego-objects*, those that contain a hidden message. Steganalysis is the opposite of *Steganography*: the body of techniques devised to hide the presence of communication. In turn, Steganography is different from *Cryptography*, that aims to make communication unintelligible for those that do not possess the correct access rights. Recently, Steganography has received a lot of attention around the world mainly because its possible applications which includes: feature location (identification of subcomponents within a data set), captioning, time-stamping, and tamper-proofing (demonstration that original contents have not been altered) [149]. Unfortunately, not all applications are harmless, and there are strong indications that Steganography has been used to spread child pornography pictures on the internet [64,113]. Hence, robust algorithms to detect the very existence of hidden messages in digital contents can help further forensic and police work. Discovering the content of the hidden message is a much more complex problem than Steganalysis, and involves solving the general problem of breaking a cryptographic code [188].

In this paper, we introduce the Progressive Randomization (PR): a new image meta-description approach suitable for different image inference applications such as broad class *Image Categorization* and *Steganalysis*. This technique captures statistical properties of the images' LSB channel, information that are invisible to the naked eye. With such perturbations, PR captures the image class separability allowing us to successfully infer high-level information about images.

The PR image meta-description approach has four stages: (1) the randomization process, that progressively perturbates the LSB value of a selected number of pixels; (2) the selection of feature regions, that makes global descriptors work locally; (3) the statistical descriptors analysis, that finds a set of measurements to describe the image; and (4) the invariance transformation, that allows us to make the descriptor's behavior image independent.

With enough training examples, PR is able to categorize images as a full self-contained classification framework. Even when only a limited number of training examples are available, the method still achieves good separability. The method also provides interesting properties for association with other image descriptors for scene reasoning purposes.

To validate our image meta-description approach for Image Categorization and Steganalysis, we have created two validation scenarios: (1) **Image Categorization**; and (2) **Hidden Messages Detection**. In the **Image Categorization** scenario, we have performed four experiments. In the first experiment, we show PR as a complete self-contained multi-class classification procedure. For that, we have used a 40,000-image database with 12,000 outdoors, 10,000 indoors, 13,500 art photographs, and 4,500 computer generated images (CGIs) with two different classification approaches: All Pairs majority voting of the binary classifier Bagging of Linear Discriminant Analysis (All-Pairs-BLDA), and SVMs [16]. In addition, we have tested the PR technique in three other categorization experiments: one to provide another interpretation of the first experiment, one for 3,354 FreeFoto images categorization into nine classes and finally, one for categorizaton of 2,950 image of fruits into 15 classes. In the **Hidden Messages Dectection** scenario, we have used the 40,000-image database first scenario to detect the very existence of hidden messages in digital images. We have used the binary classifiers: Linear Discriminant Analysis with and without Bagging ensemble, and SVMs [16].

We organize the remainder of this paper as follows. In Section 4.2, we present the Image Categorization and Steganalysis' state-of-the-art. In Section 4.3, we introduce the Progressive Randomization (PR) image meta-description approach. In Section 4.4, we validate our method for Image Categorization and, in Section 4.5, for Steganalysis, and compare our results with related work in the literature. In Section 4.6, we give a close study to the reasons of why PR works. Finally, we present the conclusions and remarks in Section 4.8.

4.2 Related work

In this section, we present recent and important achievements of Image Categorization and Steganalysis. For Image Categorization we have considered techniques from color, edge and texture properties to bag-of-features. For Steganalysis, we have considered techniques from application-specific schemes to blind detection frameworks.

4.2.1 Image Categorization

Recently, there has been a lot of activity in the area of *Image Categorization*. Previous approaches have considered patterns in color, edge and texture properties to differentiate photographs of real scenes from photographs of art [34]; low- and middle-level features integrated by a Bayesian network to distinguish indoor from outdoor images [92, 162]; histogram and DCT coefficients features to differentiate city images from landscape images [181]; and first- and higher-order wavelet statistics to distinguish photographs from photorealistic images [94]. Additional works to categorize images have considered color and texture information [111], color histograms and color correlograms [67], and border/interior pixel classification [169]

Also, there are efforts in the use of shape and silhouette [3], and moment invariants [159] to reduce the 'semantic gap' problem: images with high feature similarities may be from different categories in terms of user perception.

Fei-Fei et al. [46] have used a Bayesian approach to unsupervised one-shot learning of object categories; Oliva and Torralba [123] have proposed a computational model for scene recognition using perceptual dimensions, coined Spatial Envelope, such as naturalness, openness, roughness, expansion and ruggedness. Bosch et al [19]. have presented an unsupervised scene recognition procedure using probabilistic Latent Semantic Analysis (pLSA). Vogel and Schiele [183] have presented a semantic typicality measure for natural scene categorization.

Recent developments have used middle- and high-level information to improve the low-level features. Li et al. [84] have performed architectonics building recognition using color, orientation, and spatial features of line segments. Raghavan et al. [163] have designed a similarity-preserving space transformation method of low-level image space into a high-level vector space to improve retrieval. Some researchers have used bag of features for image categorization [62, 68, 100].

However, these approaches often require complex learning stages and can not be directly used for image retrieval tasks.

4.2.2 Digital Steganalysis

Steganography techniques can be used in medical imagery, advanced data structures designing, strong watermarking, document tracking tools, general communication, modern transit radar systems, digital elections and electronic money, document authentication, among others [149]. Unfortunately, not all applications are harmless, and there are strong indications that Steganography has been used to spread child pornography pictures on the internet [64, 113]. Hence, robust algorithms to detect the very existence of hidden messages in digital contents can help further forensic and police work.

In general, steganographic algorithms rely on the replacement of some noise component of a digital object with a pseudo-random secret message [149]. In digital images, the most common noise component is the Least Significant Bits (LSBs). To the human eye, changes in the value of the LSB are imperceptible, thus making it an ideal place for hidding information without any perceptual change in the cover object. LSB insertion/modification is considered a difficult one to detect [188].

The original LSB information may have statistical properties, so changing some of them could result in the loss of those properties. With proper statistical analysis, we can determine whether or not an image has been altered, making forgeries mathematically detectable [109]. Hence, the general purpose of Steganalysis is to collect sufficient statistical evidence about the presence of hidden messages in images, and use them to classify whether or not a given image contains a hidden content.

Westfeld and Pfitzmann [191] have introduced a powerful chi-square steganalytic technique that can detect images with secret messages that are embedded in consecutive pixels. However, their technique is not effective for raw high-color images and for messages that are randomly scattered in the image. Fridrich et al. [54] have developed a detection method based on close pairs of colors created by the embedding process. However, this approach only works when the number of colors in the images is less than 30 percent of the number of pixels. Fridrich et al [51] have analyzed the capacity for lossless data embedding in the least significant bits and how this capacity is altered when a message is embedded. It is not clear how this approach is sensible to different images given that no training stage was applied. Ker [80] have introduced a weighted least-squares steganalysis technique in order to estimate the amount of payload in a stego object. Notwithstanding, often payload estimators are subject to errors. Furthermore, their magnitude seem tightly dependent on properties of the analyzed images.

Lyu and Farid [95] have designed a classification technique that decomposes the image into quadrature mirror filters and analyzes the effect of the embedding process.

Fridrich and Pevny [134] have merged Markov and Discrete Cossine Transform features for multi-class steganalysis on JPEG images. Their approach is capable of assigning stego images to six popular steganographic algorithms. Ker [79] has introduced a new benchmark for binary steganalysis based on an asymptotic information about the presence of hidden data. The objective is to provide foundations to improve any detection method. However, there are some issues in computing benchmarks empirically and no definitive answer emerge. Rodriguez and Peterson [155] have presented an investigation of using Expectation Maximization for hidden messages detection. The contribution of their approach is to use a clustering stage to improve detection descriptors.

4.3 Progressive Randomization approach (PR)

Small perturbations in the LSB channel are imperceptible to humans [188] but are statistically detectable for image analysis. Here, we introduce the Progressive Randomization image meta-description approach for *Image Categorization* and *Steganalysis*. It is a new image meta-description approach that captures the differences between broad-image classes using the statistical artifacts inserted during the perturbation process. Our experiments in Sections 4.4 and 4.5 demonstrate PR captures the image class separability allowing us to successfully infer high-level information about images.

Algorithm 5 summarizes the four stages of PR applied to Image Categorization and Steganalysis: (1) the randomization process (c.f., Sec. 4.3.2); (2) the selection of feature regions (c.f. Sec. 4.3.3); (3) the statistical descriptors analysis (c.f. Sec. 4.3.4); and (4) the invariance transformation (c.f. Sec. 4.3.5).

In summary, if we use $n = 6$ controlled transformations, we have to analyze the perturbation artifacts in seven images (the input plus the perturbed ones). Furthermore, if we analyze $r = 8$ regions per image and use $m = 2$ statistical descriptors for each region, we have to assess $r \times m = 16$ features for each image. In this context, the final PR description vector amounts to $(n + 1) \times r \times m = 112$ features. If we perform the last step of invariance (which depends on the application), we normalize each group of features of one perturbed image with respect to the feature values in the input image. Therefore, the final description vector after normalization amounts to $n \times r \times m = 96$ features.

4.3.1 Pixel perturbation

Let $\mathbf{x}$ be a Bernoulli distributed random variable with $Prob\{\mathbf{x} = 0\}) = Prob(\{\mathbf{x} = 1\}) = \frac{1}{2}$, B be a sequence of bits composed by independent trials of $\mathbf{x}$, p be a percentage, and S be a random set of pixels of an input image.

Given an input image I of $|I|$ pixels, we define the LSB pixel perturbation $T(I, p)$ the process of substitution of the LSBs of S of size $p \times |I|$ according to the bit sequence B. Consider a pixel $px_i \in S$ and an associated bit $b_i \in B$

$$\mathcal{L}(px_i) \leftarrow b_i \text{ for all } px_i \in S. \tag{4.2}$$

where $\mathcal{L}(px_i)$ is the LSB of the pixel px_i.

Figure 4.1 shows an example of a perturbation using the bits $B = 1110$.

Algorithm 5 The PR image meta-description approach

```
Require: Input image $I$; Percentages $P = \{P_1, \ldots P_n\}$;
Randomization: perform $n$ LSB pixel disturbances of the input image    ▷ Sec. 4.3.2
    $\{O_i\}_{i=0 \ldots n.} = \{I, T(I, P_1), \ldots, T(I, P_n)\}.$
Region selection: select $r$ feature regions of each image $i \in \{O_i\}_{i=0 \ldots n}$    ▷ Sec. 4.3.3
    $\{O_{ij}\}_{\substack{i = 0 \ldots n, \\ j = 1 \ldots r.}} = \{O_{01}, \ldots, O_{nr}\}.$
Statistical descriptors: calculate $m$ descriptors for each region    ▷ Sec. 4.3.4
    $\{d_{ijk}\} = \{d_k(O_{ij})\}_{\substack{i = 0 \ldots n, \\ j = 1 \ldots r, \\ k = 1 \ldots m.}}$
Invariance: normalize the descriptors based on their behavior in the input image $I$    ▷ Sec. 4.3.5
    $\mathbf{F} = \{f_e\}_{e=1 \ldots n \times r \times m} = \left\{ \frac{d_{ijk}}{d_{0jk}} \right\}_{\substack{i = 0 \ldots n, \\ j = 1 \ldots r, \\ k = 1 \ldots m.}},$    (4.1)
Use $\{d_{ijk}\} \in \mathbb{R}^{(n+1) \times r \times m}$ (non-normalized) or $\{d_{ijk}\} \in \mathbb{R}^{n \times r \times m}$ (normalized) features in your favorite machine learning black box.
```

4.3.2 The randomization process

Given an input image I, the randomization process consists in the progressive application $I, T(I, P_1), \ldots, T(I, P_n)$ of LSB pixel disturbances. The process returns n images that only differ in the LSB from the input image, and are identical to the naked eye.

The $T(I, P_i)$ transformations are perturbations of different percentages of the available LSBs. Here, we use $n = 6$ where $P = \{1\%, 5\%, 10\%, 25\%, 50\%, 75\%\}$, $P_i \in P$ denotes the relative sizes of the set of selected pixels S. The greater the LSB pixel disturbance, the greater the resulting LSB entropy of the transformation. Figure 4.2 shows that the $T(I, P_i)$ transformations does not introduce visual changes in the images. The perturbations are performed only over the LSBs of the input image (Figure 4.3). Hence, the differences between the input image and any other perturbed image are in the LSB channel only.

4.3.3 Feature region selection

Local image properties do not show up under a global analysis [188]. We use statistical descriptors on local regions to capture the changing dynamics of the statistical artifacts inserted during the randomization process (c.f., Sec. 4.3.2).

Given an image I, we use r regions with size $l \times l$ pixels to produce localized statistical descriptors. In Figure 4.4, we show the $m = 8$ overlapping regions we use in this paper.

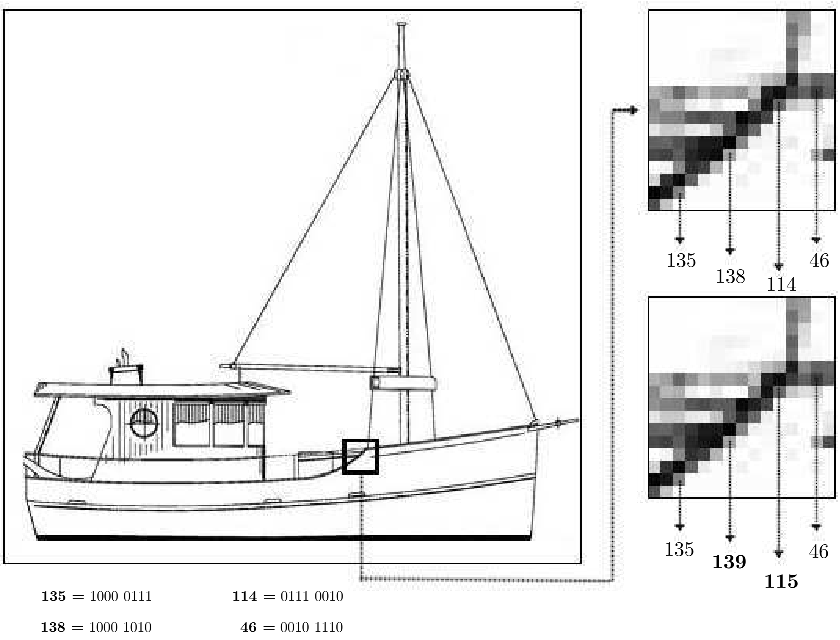


Figure 4.1: An example of LSB perturbation using the bits $B = 1110$.

The curse of dimensionality keeps us from adding too many regions — we have found out, experimentally, that eight regions are a good tradeoff.

4.3.4 Statistical descriptors

The LSB perturbation procedure changes the contents of a selected number of pixels and induces local changes of pixel statistics. An L-bit pixel spans 2^L possible values, and has 2^{L-1} classes of invariance under pixel perturbations (c.f., Sec. 4.3.1). Let's call these invariant classes *pair of values* (PoV).

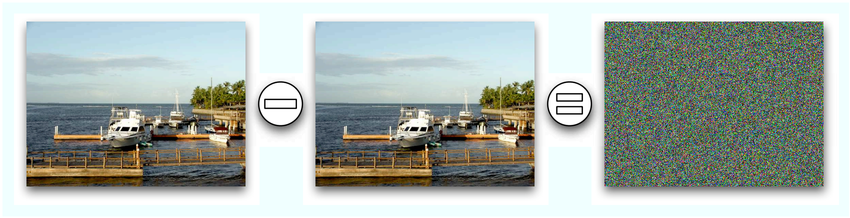


Figure 4.2: The input image I (leftmost) and its perturbed version $T(I, 75\%)$ (middle): differences are not visible by the naked eye. However, they are present in the LSB channel (rightmost).

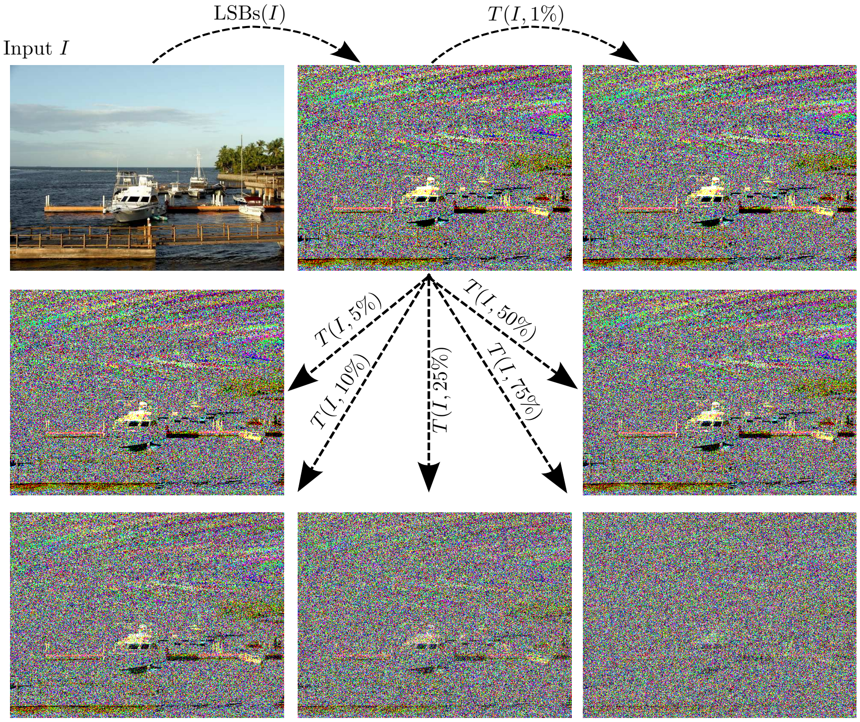


Figure 4.3: The Progressive Randomization behavior over the LSBs. $T(I, P_i)$ represents a PR perturbation with percentage P_i over the LSBs of the input image I.

When we disturb all the available LSBs in S with a sequence B, the distribution of $0/1$ values of a PoV will be the same as in B. The statistical analysis compares the theoretically expected frequency distribution of the PoVs with the observed ones after the perturbation process.

We apply the χ^2 (chi-squared test) [191] and U_T (Ueli Maurer Universal Test) [102] to analyze the images.

χ^2 test

The χ^2 test [48] compares two histograms f^{obs} and f^{exp}. Histogram f^{obs} represents the observations and f^{exp} represents the expected histogram. The procedure computes the sum of the square differences of f^{obs} and f^{exp} divided by f^{exp},

$$\chi^2 = \sum_i \frac{(f_i^{obs} - f_i^{exp})^2}{f_i^{exp}}. \tag{4.3}$$

Ueli test

The Ueli test (U_T) [102] is an effective way to evaluate the randomness of a given sequence of numbers. U_T splits an input data S into n blocks. For each block b_i, it analyzes each of the $n-1$

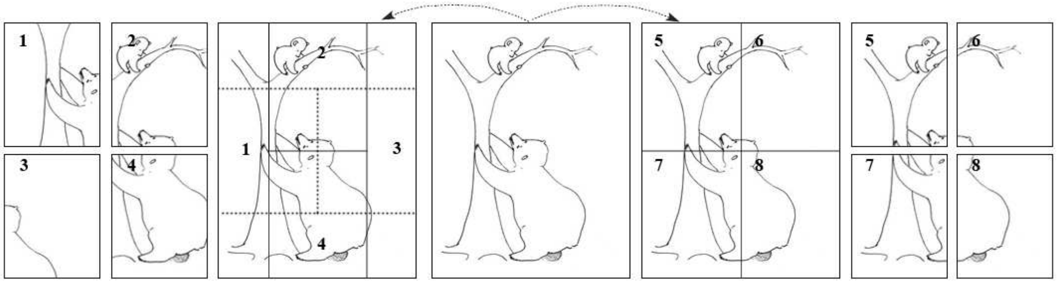


Figure 4.4: The eight overlapping regions used in the experiments.

remaining blocks, looks for the most recent occurrence of b_i, and takes the log of the summed temporal occurrences. Let $B(S) = (b_1, b_2, \ldots, b_N)$ be a set of n blocks such that $\cup_{\forall b_i} = S$. Let $|b_i| = L$ be the block size for each i and $|B(S)| = N$ be the number of blocks. We define $U_T : B(S) \rightarrow \mathbb{R}^+$ as a function

$$U_T(B(S)) = \frac{1}{K}\sum_{i=Q}^{Q+K} \ln A(b_i), \tag{4.4}$$

where K is the number of analyzed bits (e.g., $K = N$), Q is a shift in $B(S)$ (e.g., $Q = \frac{K}{10}$ [102]), and

$$A(b_i) = \begin{cases} i & \nexists i' \in \mathbb{N}, i' < i \rightarrow b_{i'} = b_i, \\ \min\{i' : b_{i'} = b_i\} & \text{otherwise.} \end{cases}$$

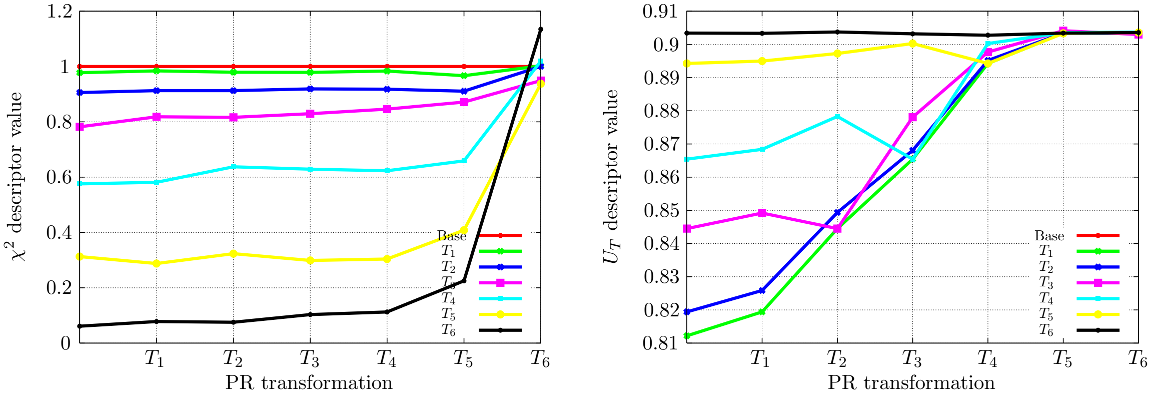
In practice, if U_T is close to 7.1836, we have a high randomness condition. On the other hand, the lower U_T, the more predictable is the condition in S.

4.3.5 Invariance

In some situations, it is necessary to use an image-invariant feature vector. For that, we use the variation rate of our statistical descriptors with regard to the PR, rather than their values. We normalize all descriptors from the transformations with regard to their values in the input image

$$F = \{f_e\}_{e=1\ldots n\times r\times m} = \left\{\frac{d_{ijk}}{d_{0jk}}\right\}_{\substack{i=0\ldots n,\\ j=1\ldots r,\\ k=1\ldots m.}}, \tag{4.5}$$

where d denotes a descriptor $1 \leq k \leq m$ of a region $1 \leq j \leq r$ of an image $0 \leq i \leq n$ and F is the final generated descriptor vector of the image I. Figures 4.5(a-b) show the behavior of our statistical descriptors along the progressive randomization of one selected image I.



(a) χ^2 descriptor.

(b) U_T descriptor.

Figure 4.5: Normalized descriptor's behavior along the progressive randomization. T_i represents the PR operation $T(I, P_i)$.

The need for invariance depends on the application. For instance, it is necessary for Steganalysis but harmful for Image Categorization. In Steganalysis, we want to differentiate images that do not contain hidden messages from those that contain hidden messages, and the image class is not important. On the other hand, in Image Categorization, the descriptor values are important to improve the class differentiation. Different classes do have distinct behavior under Progressive Randomization approach (c.f., Sec. 4.4, 4.5, and 4.6).

4.4 Experiments and results – Image Categorization

In this section, we describe how we train, test and validate PR image meta-description approach for *Image Categorization*. We validate the multi-class classification as a complete self-contained classification procedure in **Experiment 1**. In that experiment, we use a 40,000-image database with 12,000 outdoors, 10,000 indoors, 13,500 art photographs, and 4,500 computer generated images (CGIs) with three different classification approaches.

The images in **Experiment 1** come from five main sources: Mark Harden's Artchive[1], the European Web Gallery of Art[2], FreeFoto[3][4], Berkeley CalPhotos[5], and from The Internet Ray Tracing Competition (IRTC)[6]. Figure 4.6 show some examples of each category.

We also validate the PR image meta-description approach in three other categorization

[1] http://www.artchive.com

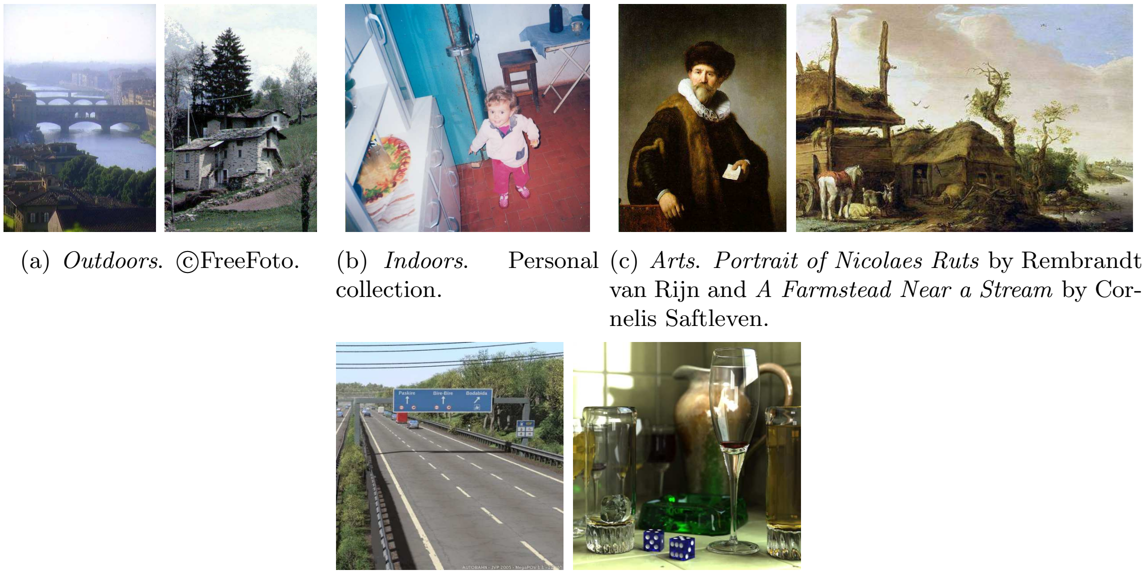
[2] http://www.wga.hu

[3] http://www.freefoto.com

[4] http://www.ic.unicamp.br/~rocha/pub/communications.html

[5] http://calphotos.berkeley.edu

[6] http://www.irtc.org



(a) *Outdoors.* ©FreeFoto.

(b) *Indoors.* Personal collection.

(c) *Arts. Portrait of Nicolaes Ruts* by Rembrandt van Rijn and *A Farmstead Near a Stream* by Cornelis Saftleven.

(d) *Computer Generated Images. Autoban* by Jaime Vives Piqueres and *Glasses* by Gilles Tran.

Figure 4.6: Examples of each analyzed category: Outdoors, Indoors, CGIs, and Arts.

scenarios. In **Experiment 2**, we provide another interpretation of **Experiment 1** using a second carefully assembled image database. In **Experiment 3**, we perform a 9-class image categorization using 3,354 FreeFoto photographs. Finally, **Experiment 4**, we perform a 15-class image categorization using 2,950 images of fruits.

4.4.1 Experiment 1

We compare our method to the state of the art two-class separation approaches in the literature [34, 92, 94, 128, 162] using a simple *Bagging ensemble* of Linear Discriminant Analysis (BLDA) [55]. Furthermore, we also perform multi-class image-categorization, separating *Outdoor photographs*, *Art images*, *Photorealistic Computer Generated Images*, and *Indoors photographs*. Here, we use the BLDA classifier with 13 iterations and 10-fold cross validation.

Two-class classification

Cutzu et al. [34] have addressed the problem of differentiating photographs of real scenes from photographs of art works. They validated over a database with 6,000 photographs from FreeFoto and 6,000 photographs from Mark Harden's Artchive and from Indiana Image Collection[7].

The authors have used color and intensity edges, color variation, saturation, and Gabor features in a complex classifier. We use a similar image set reported in [34]. We have selected

[7] http://www.dlib.indiana.edu/collections/dido

12,000 photographs and 13,500 art photographs totalizing 25,500 images.

Lyu and Farid [94] have used a statistical model based on first- and higher-order wavelet statistics to reveal significant differences of photographs and photorealistic images. They have used photographs from FreeFoto and photorealistic images from IRTC and from Raph 3D Artists.

We use almost the same image set reported in [94]. Therefore, we have used only images from FreeFoto, IRTC and Raph sources, 7,500 photographs and 4,700 photorealistic images, totalizing 12,200.

Luo and Savakis [92, 162] have associated texture and color information about sky and grass to differentiate indoors and outdoors images. They have used a Kodak image database not freely available. Payne and Singh [128] have used edge informations to differentiate indoors from outdoors images in a personal image collection.

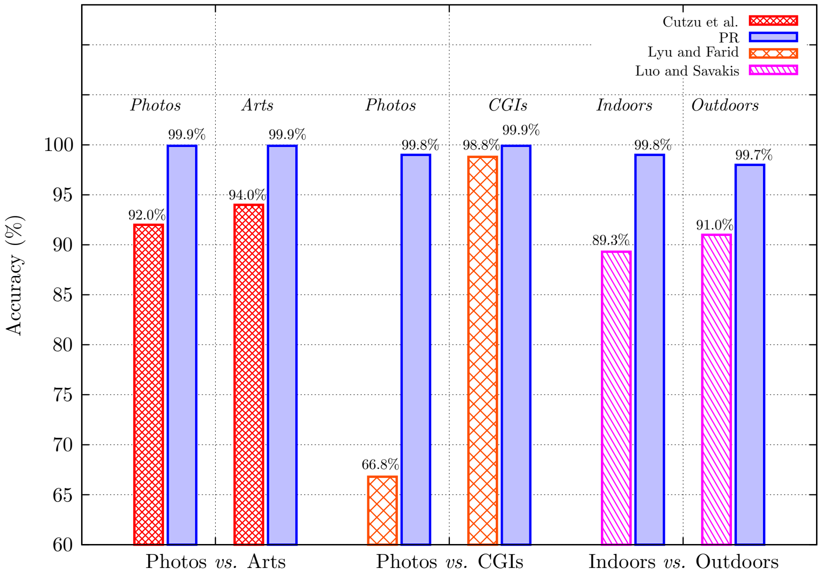


Figure 4.7: **Experiment 1**. PR description approach used to binary *Image Categorization* using 10-fold cross- validation.

PR distinguishes *Photographs* from *Art* images with an average accuracy of $\frac{\mu_1+\mu_2}{2} = 99.9.\%$, *photographs* from *CGI* images with an average accuracy of $\frac{\mu_1+\mu_2}{2} = 99.9\%$ and *Indoors* from *Outdoors* images with an average accuracy of $\frac{\mu_1+\mu_2}{2} = 99.7\%$.

Multi-class classification

The PR approach creates a single descriptor that works for different image inference applications. For instance, PR is suitable for multi-class broad image categorization such as the four classes *Indoors*, *Outdoors*, *CGIs*, and *Arts*.

In order to validate the multi-class classification, we have used two different approaches that are combinations of binary classifiers: All Pairs majority voting of the binary classifier BLDA (All-Pairs-BLDA); and Support Vector Machines (SVMs). LibSVM uses an internal mechanism that put together all 1 × 1 combinations of the classes and performs a majority voting in the final stage. We have used the radial basis function SVM. All-Pairs-BLDA uses sets of binary classifiers. Note that, any other binary classifier could be used rather than All-Pairs-BLDA.

Tables 4.1 and 4.2 show the resulting classification using All-Pairs-BLDA, and SVMs. The diagonal represents the classification accuracy. For instance, using All-Pairs-BLDA multi-class approach 89.4%, of the images that represent an *Art* scenario are correctly classified, while only 8.17% of them are misclassified as *Indoors*.

	All-Pairs-BLDA Predictions			
	Arts	**CGIs**	**Indoors**	**Outdoors**
Arts	89.4% ± 1.04%	4.41% ± 0.49%	6.16% ± 0.80%	0.00% ± 0.00%
CGIs	33.66% ± 2.36%	53.3% ± 2.09%	13.0% ± 1.22%	0.00% ± 0.00%
Indoors	8.97% ± 0.54%	5.58% ± 0.34%	85.44% ± 0.62%	0.01% ± 0.03%
Outdoors	0.00% ± 0.00%	0.06% ± 0.07%	0.02% ± 0.05%	99.9% ± 0.11%

Table 4.1: **Experiment 1**. PR multi-class *Image Categorization* using All-Pairs-BLDA.

The PR approach is independent of the multi-class technique. Figure 4.8 shows the minimum accuracy of the two approaches as well as the average accuracy and the geometric average accuracy.

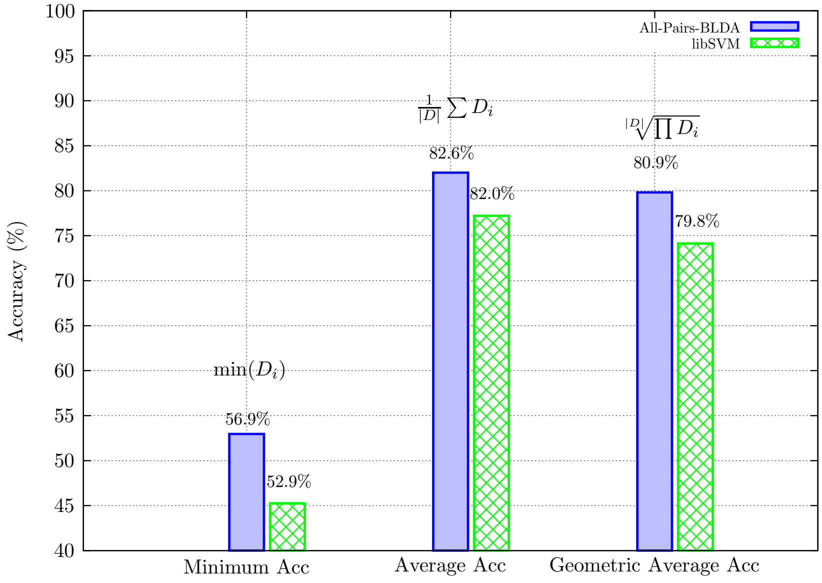


Figure 4.8: **Experiment 1**. Multi-class overall accuracy. D is the diagonal of the confusion matrix.

	SVM Predictions			
	Arts	**CGIs**	**Indoors**	**Outdoors**
Arts	86.4% ± 1.16%	2.10% ± 0.43%	11.5% ± 0.85%	0.00% ± 0.00%
CGIs	36.2% ± 2.32%	45.3% ± 1.55%	18.2% ± 1.34%	0.20% ± 0.20%
Indoors	17.5% ± 1.28%	4.90% ± 0.41%	77.6% ± 1.47%	0.00% ± 0.00%
Outdoors	0.02% ± 0.03%	0.37% ± 0.18%	0.01% ± 0.03%	99.6% ± 0.21%

Table 4.2: **Experiment 1**. PR multi-class *Image Categorization* using SVM.

4.4.2 Experiment 2

Experiment 1 provides good performance in two-class and multi-class categorization. However, some can argue that the number of training examples is too large and that it might have suffered from bias due to different compression applied to each category, given that they come from different sources. It is important to notice that almost all images come from well known image repositories and most of them are built up from user contributions.

So, we have created a second scenario for multi-class categorization of *Indoors*, *Outdoors*, *CGIs* and *Art photographs*. In this experiment, we have manually selected 500 images for each class totalizing 2,000 images. Each category contains images from at least 75 different internet sources and there are no more than seven images from the same place. There is no intersection among the images in this scenario with the scenario presented in **Experiment 1**. In this experiment, there are at least 400 different cameras with many different compression scenarios. Figure 4.9 presents the results using the All-Pairs BLDA multi-class approach with 13 iterations.

We show that the results using PR descriptor are not biased due to possible different compression levels. The results are better than the priors of each class (about 25% per class). Therefore, the PR descriptor provides good separation among classes. Even with few training examples, the descriptor still presents a good performance. The more examples we provide in the training phase, the better the classification performance (Figure 4.9).

4.4.3 Experiment 3

Here, we select 3,354 images from FreeFoto and divide them into nine classes. Figure 4.10 shows some examples of each category. *Sky and Clouds* category represents sunny and clear days. *Cummulonimbus Clouds* comprises images associated with heavy precipitation and thunderstorms. The other categories are self explanatory.

We do not pre-process any image. All images come from FreeFoto and were originally stored in JPEG format with 72 DPIs using similar compression levels. Figure 4.11 presents the results for this experiment using the All-Pairs BLDA multi-class approach with 13 iterations.

The PR meta-description approach generalyzes from the priors (about $\frac{1}{9}$ for each category). The accuracy increases with the number of training examples (left plot of Figure 4.11). The more images in the training phase, the more accurate the classification. This suggests that PR technique can be combined with other image descriptors for categorization purposes. The average standard deviation σ is bellow 5%. For all classes, the classification results are far above

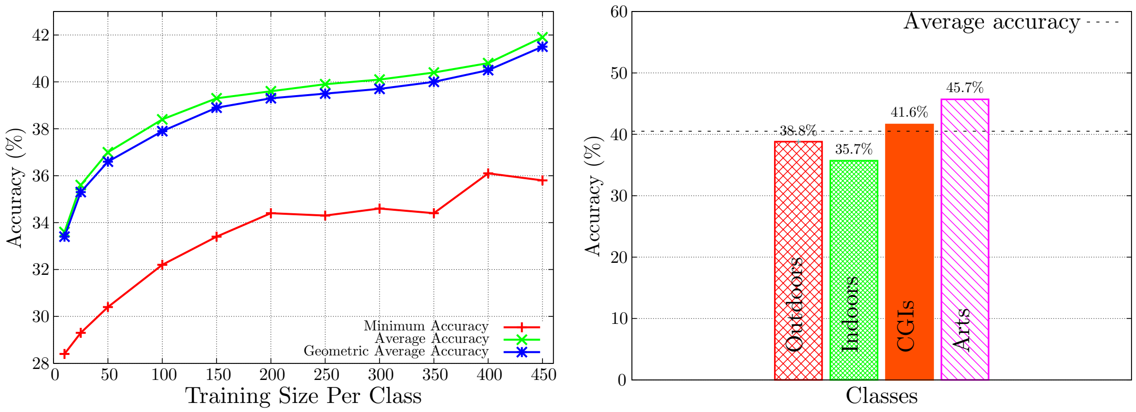


Figure 4.9: **Experiment 2**. Out × Ind × CGI × Arts using All-Pairs BLDA(13) .:. 4 classes. *Left plot*: average performance for variable training sizes. *Right plot*: class' performance for 450-sized training sets.

the expected priors (2σ minimum).

4.4.4 Experiment 4

In this experiment, we perform categorization of images of fruits[8] and we want to show that the PR results are not biased due to camera properties. Here, we have used the same camera and setup in the capture. The JPEG compression level is the same for all images.

We personally acquired the 2,950 images at our local fruits and vegetables distribution center (CEASA), using a Canon PowerShot P1 camera, at a resolution of $1,024 \times 768$ against a white background. Figure 4.12 depicts the 15 different classes. Even in the same category, there are many illumination differences (Figure 4.13).

Figure 4.14 presents the results for this experiment using the All-Pairs BLDA multi-class approach with 13 iterations. Clearly, PR generalyzes from the priors (about $\frac{1}{15}$ for each category) and the accuracy increases with the number of training examples (left plot of Figure 4.14).

4.5 Experiments and results – Steganalysis

In this section, we describe how we train, test and validate the PR image meta-description approach for *Steganalysis*. In this scenario, our objective is to detect whether or not a given image contains an embedded content. Here, we have used the same image database of Experiment 1 in Section 4.4.1.

[8] `http://www.ic.unicamp.br/~rocha/pub/communications.html`

Trees / Forests
Ocean
Sky and Clouds
Flowers
Leaves
Sunrise / Sunset
Cummulonimbus
Snow Mountains
Water Streams

Figure 4.10: Nine FreeFoto categories.

Among all message embedding techniques, the *Least Significant Bit* (LSB) insertion/modification is considered a difficult one to detect [149, 188]. In general, it is enough to detect whether a message is hidden in a digital content. For example, law enforcement agencies can track access logs of hidden contents to build a network graph of suspects. Later, using other techniques, such as physical inspection of apprehended material, they can uncover the actual contents and apprehend the guilty parties [73, 149].

4.5.1 Overall results

We define a stego-image as an image that suffered an LSB pixel disturbance. The amount of disturbance inserted using the sequence of bits B represents the size of a possible information (message) that is embedded $|M|$. We train a classifier with stego and non-stego examples. To obtain stego examples, we simulate message embeddings perturbing the LSBs of an image subset of our database. We have created a version of our image database for each one of our selected content-hiding scenarios (relative size of contents to the embedding capacity of the image).

In Figure 4.15, we present the overall results for the PR technique applied for hidden messages detection. We obtain the best results when using the Bagging Ensemble with Linear Discriminant Analysis (BLDA). For instance, for a relative-size message embedding of 10%, PR yields 78.1% of accuracy. That is almost the same result that the more computationally intensive procedure of SVM. Furthermore, it worth noting that SVM does not benefit from the Bagging

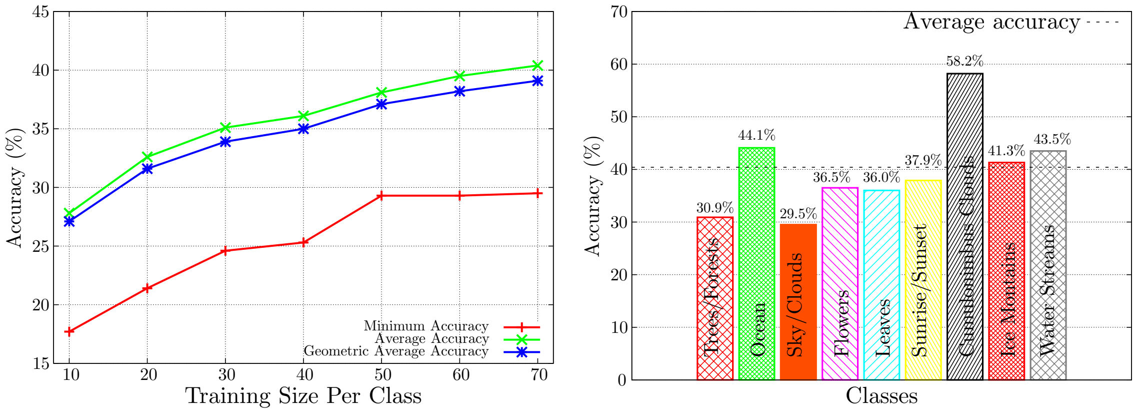


Figure 4.11: **Experiment 3**. FreeFoto categorization using All-Pairs BLDA(13) .:. 9 classes. *Left plot*: average performance for variable training sizes. *Right plot*: class' performance for 70-sized training sets.

ensemble. That is because SVM uses only the elements close to the margins in the classification procedure.

PR descriptor scales with the number of examples in the training stage. Overall, the more examples we provide in the training phase, the greater the detection accuracy regardless the message size. In Figure 4.16, we present the PR descriptor with different training set sizes.

The detection of very small relative-size contents is very hard, and still an open problem. Nevertheless, in practical situations, like when pornographers use images to sell their child-porn images, they usually use a reasonable portion of the LSB channel available space (e.g., 25%). In this class of problem, PR meta-description approach detects such activities with accuracy just under 90% which shows that it is an effective approach for embedding content detection.

4.5.2 Class-based Steganalysis

Different classes/categories of images have a very distinct behavior in properties. We explored their different LSB behavior earlier in this paper for proper image categorization.

In this section, we show how the PR descriptor is still able to perform Steganalysis despite all these differences, and gives us a strong insight about which types of images are better for information hiding.

We have found that the detection of hidden content in images with low richness of detail (e.g., *Indoors*) is easier. The inserted artifacts of the embedding process in these images are more obvious than those artifacts inserted in images with more complex details. In these experiment, we have considered four image classes *Outdoors*, *Indoors*, *Arts*, and *CGIs*. We have used the same image database of Experiment 1 in Section 4.4.1.

In each analysis, we train our classifier with examples sampled without replacement from

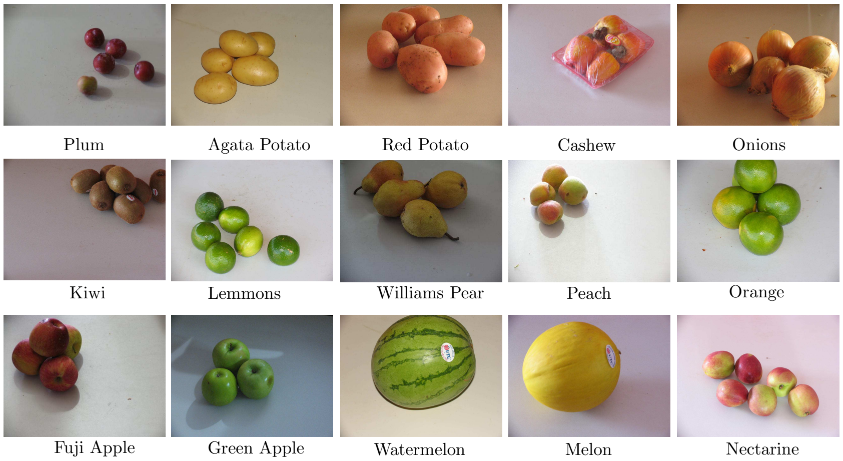


Figure 4.12: Fifteen categories of fruits.

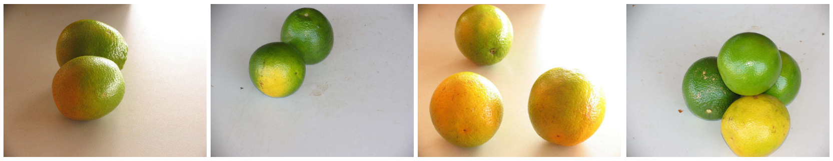


Figure 4.13: Illumination differences in the *Orange* class.

three classes, and test in the fourth class. We repeat the process to test each class. Figure 4.17 shows the resulting classification accuracy for each class of image. We also show the expected classification value when we train and test over all classes with a proportion of 70% examples for training and 30% for testing.

Looking at the results, we conclude that the classes *Arts* and *Outdoors* are the most difficult types to detect hidden messages. On the other hand, *Indoors* images are the easier ones to detect hidden messages. Finally, as our intuition would expect, the greater the message, the better the classification accuracy no matter the class.

4.5.3 Comparison

Westfeld and Pfitzmann [191] have devised an approach that only detects sequential hidden messages embedded from the first available LSB. This approach is not robust to image variability and it is not able to detect messages altered from some embedding message procedure that preserves statistics such as mean, and variance about the cover-image. Our framework overcomes these problems and increases the classification accuracy. We compare the results in Table 4.3.

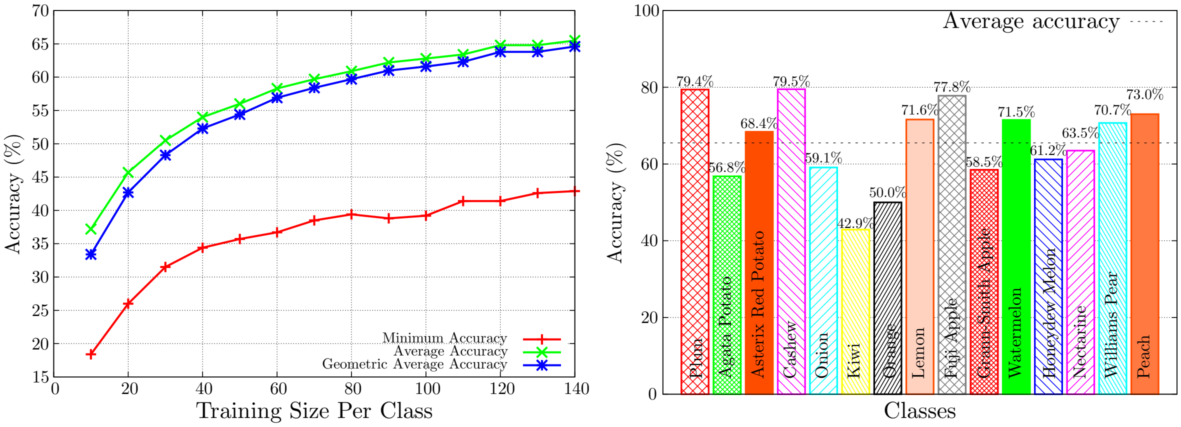


Figure 4.14: **Experiment 4**. Fruits categorization using All-Pairs BLDA(13) .:. 15 classes. *Left plot*: average performance for variable training sizes. *Right plot*: class' performance for 140-sized training sets.

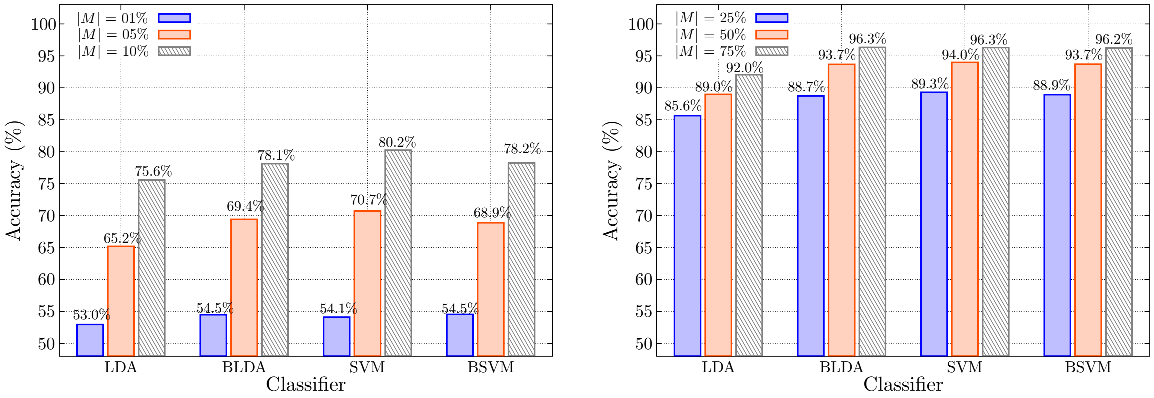
In this experiment, we have used the SVM binary classifier [16].

Other approaches for Steganalysis include the works of Shi et al. [168], Pevny and Fridrich [134], and Goljan et al. [59]. However, these approaches have been designed for embedding techniques based on *lossy compression* properties such as those present in JPEG images. In this paper, we present a detection framework designed for *lossless* embedding detection.

It is not intended that PR outperforms all the best Steganalysis algorithms in the literature. It is worth noting that we are not providing a complete Steganalysis framework. Indeed, in this paper, we present a new image meta-description approach that can be used for Steganalysis. For this reason, we compare our descriptor with two well-known state-of-the-art solutions. PR association with other image descriptors is straightforward.

	WP	**PR**
	$\mu \pm \sigma$	$\mu \pm \sigma$
01%	52.6% ± 0.1%	54.1% ± 0.9%
05%	52.6% ± 0.1%	70.7% ± 0.9%
10%	54.6% ± 4.1%	80.2% ± 0.5%
25%	72.9% ± 1.9%	89.3% ± 0.6%
50%	83.0% ± 0.6%	94.0% ± 0.5%
75%	84.8% ± 0.9%	96.3% ± 0.3%

Table 4.3: Westfeld and Pfitzmann's detection approach (WP) *vs.* Progressive Randomization (PR). μ and σ from cross-validation.



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

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

Figure 4.15: PR accuracy for different message embeddings scenarios.

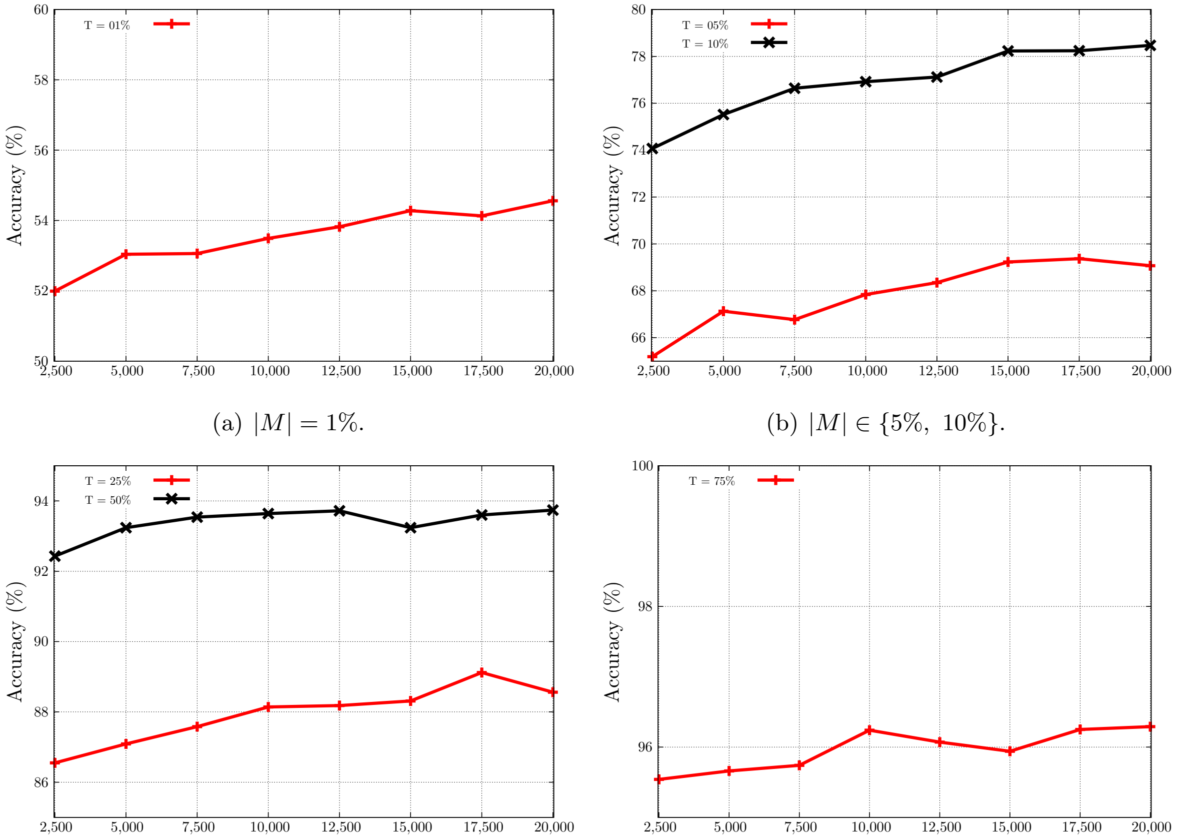
Our results are about 26 percentage points better than Westfeld-Pfitzmann's results for small relative-size message embeddings (e.g., $|M| = 10\%$) and are about 17 percentage points better than Westfeld-Pfitzmann's results for medium relative-size message embeddings (e.g., $|M| = 25\%$).

Lyu and Farid [95] have designed a technique that decomposes the image into quadrature mirror filters to analyze the effect of the embedding process. They have used a database of about 40,000 images. The authors tuned their classifiers parameters to have a false positive rate of only 1%. We compare the results in Table 4.4. The accuracy showed there, for comparison, is the percentage of the stego-images correctly classified. Our Progressive Randomization descriptor detects small (e.g., $|M| = 10\%$) and medium (e.g., $|M| = 50\%$) relative-size message embeddings with an accuracy of about nine percentage points better than Lyu and Farid's approach. When we consider large relative-size message embeddings (e.g., $|M| = 99\%$), our descriptor is about 19 percentage points (about 31 standard deviations) better than Lyu and Farid's approach.

4.6 Why does PR work?

We initially conceived the Progressive Randomization for Steganalysis of LSB hiding techniques [147]. In this image reasoning scenario, the behavior of the randomization steps is clear: each step emulates hiding a message with a different size. This process is conceptually similar to deciding whether the data is already compressed by looking at the statistics of a new compression operation over this data.

The experiments in Section 4.4 have demonstrated that PR captures the image class separability allowing us to successfully categorize images. However, the successive-compressions



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

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

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

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

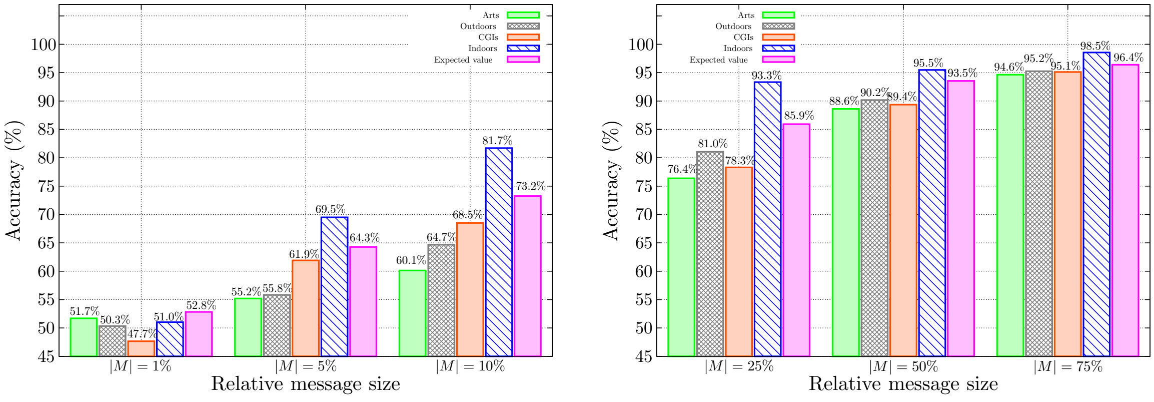
Figure 4.16: PR accuracy for different training set sizes and stego scenarios.

analogy, so intuitive in Steganalysis, is not convincing for this new problem. Our conjecture is that the distinct class behavior comes from the interaction between different light spectrum and the sensors during the acquisition process. That supports that fact that *Outdoors* is easier to differentiate from the other classes, and that *Indoors* and *Arts* are harder to differentiate amongst each other, as both use artificial illumination.

To show that the separability is not due to different patterns of luminance/color amongst classes, we have devised an experiment to measure the expected value of the Ueli descriptor conditioned to the luminance of the region.

We use a local sliding window to calculate local luminance and Ueli, and compute them on all possible 32×32-pixel windows in 300 examples of each class to estimate the $E[U_T|Lum, Class]$, the conditional expectation of Ueli for given luminance and class.

We approximate the continuous function using histograms of expected values of U_T for each



(a) (b)

Figure 4.17: PR Steganalysis along with different image classes and different relative message sizes.

class H_i^E, $i \in$ {Outdoor, Indoor}

$$H_i^E \leftarrow E[Ueli|L \in 1 \ldots 255], \tag{4.6}$$

where $E[.]$ is the statistical expectation, and L is the luminance such that $L = (0.3 * R) + (0.59 * G) + (0.11 * B)$.

The upper plot of Figure 4.18 displays the Ueli conditional expectation for unmodified *Outdoors* and *Indoors* classes. There is a consistent difference between classes, showing that the separability of the LSB statistical descriptors is not due to different class patterns of luminance.

We also observe the effect of the limited dynamic range on the statistical descriptor. Luminance components that are too small are squished to zero, while color components that should be very high are clamped to the maximum (255 in the 8-bit case). In these extreme cases, there is no randomness, and the Ueli value goes down to zero. As we calculate the expected values in sliding windows, the decrease along the borders of the dynamic range demonstrate the decrease of the randomness as more elements of the window are clamped to an extreme value.

4.7 Limitations

In this paper, we have introduced a new image meta-description approach. Its main difference with respect to the state-of-the-art techniques is that it is based on controlled perturbations on least significant properties of images.

PR technique is not intended to be the final word for Image Categorization and Steganalysis on its own. Of course, there are some limitations with the method. First of all, it is more suitable for loss-less images or high-quality lossy-compressed images, i.e., if one performs a medium-to-

	LDA		**SVM-RBF**		**Type**
	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

Table 4.4: Lyu and Farid's detection approach (LF) *vs.* Progressive Randomization (PR) considering FPR = 1%. μ and σ from cross-validation. Lyu and Farid's results from [95].

high-rate lossy compression (e.g., +25% of loss), it is possible that the method will fail. However, there are a lot of available images in the internet with low-to-medium compression levels.

In the case of Steganalysis, if one destroys the LSB channel information using a pseudo-random number generator (PRNG), the method potentially will fail. However, in this case, even the message will be destroyed.

Finally, PR technique probably will fail when used for Image Categorization of images acquired with old cameras with low-quality capturing sensors. In such situations, it is likely that the LSB channel information is related to noise in the process of acquisition. Hence, the relationship of the LSB channel with the other bit channels becomes weaker.

4.8 Conclusions and remarks

We have introduced a new image descriptor descriptor that captures the changing dynamics of the statistical artifacts inserted during a perturbation process in each of the broad-image classes of our interest.

We have applied and validated the Progressive Randomization descriptor in two real image inference scenarios: *Image Categorization* and *Steganalysis*.

The main difference among PR and the state-of-the-art algorithms is that it is based on perturbations on the values of the *Least Significant Bits* of images. With such perturbations, PR captures the image class separability allowing us to successfully infer high-level information about images. Our conjecture is that the interaction of different light spectrum with the camera sensors induces different patterns in the LSB field. PR does not consider semantical information about scenes.

The most important features in the PR descriptor are its low dimensionality and its unified approach for different applications (e.g., the class of an image, the class of an object in a restricted

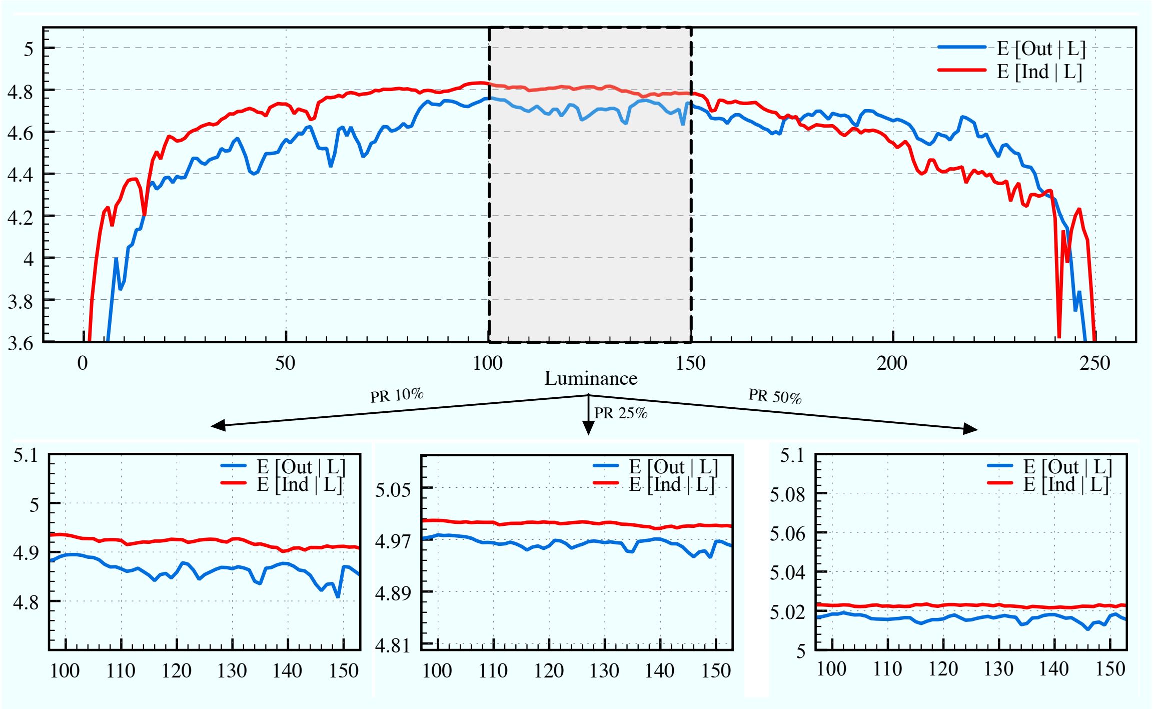


Figure 4.18: Dynamic ranges of the conditional Ueli descriptor given the luminance variation. Original image set, and the same image set with perturbations of 10%, 25%, and 50%, respectively. The plots are in different scales.

domain, hidden messages detection) even with different cameras and illumination.

With enough training examples, PR on its own is able to categorize images as a full self-contained classification framework. However, huge training sets are not always available. When only a limited number of training examples are available, the method still achieves good separability, and its accuracy increases with the size of the training set.

We have demonstrated that PR approach can perform Steganalysis despite differences in the image classes, giving us a strong insight about which types of images are better for information hiding. As our intuition would expect, the greater the message, the better the classification accuracy no matter the class. The detection of very small relative-size contents is very hard, and still an open problem. Nevertheless, in practical situations, like when pornographers use images to sell their child-porn images, they usually have to use a reasonable portion of the LSB channel available space (e.g., 25%). In this class of problem, our approach detects such activities with accuracy just under 90% which shows its effectiveness to hidden content detection.

PR descriptor presents two interesting properties that indicate that it can be combined with other image descriptors such as those described earlier in this paper. First, it generalyzes from the priors even for small training sets. Second, the accuracy increases with the number of training examples in all applications we have showed.

Future work include: to select image regions rich in details and analyze how they are affected using PR descriptor; to investigate other statistical descriptors besides χ^2 and U_T such as kurtosis and skewness; and to apply PR descriptor to other image inference scenarios such as image forgery detection.

Fusão Multi-classe de Características e Classificadores

Algumas vezes, problemas de categorização multi-classe são complexos e a fusão de informações de vários descritores torna-se importante.

Embora a fusão de características seja bastante eficaz para alguns problemas, ela pode produzir resultados inesperados quando as diferentes características não estão normalizadas e preparadas de forma adequada. Além disso, esse tipo de combinação tem a desvantagem de aumentar o número de características do vetor base de descrição o que, por sua vez, pode levar à necessidade de mais elementos para o treinamento.

No Capítulo 5, nós apresentamos uma abordagem para combinar classificadores e características capaz de lidar com a maior parte dos problemas citados anteriormente. Nosso objetivo é combinar um conjunto de características e os classificadores mais apropriados para cada uma de modo a melhorar a performance sem comprometer a eficiência.

Nós propomos lidar com um problema multi-classe a partir da combinação de um conjunto de classificadores binários. Nós validamos nossa abordagem numa aplicação real para classificação automática de frutas e legumes.

O trabalho apresentado no Capítulo 5 é uma compilação de nosso artigo submetido à *Elsevier Computers and Electronics in Agriculture* (Compag). Os autores desse artigo, em ordem, são: Anderson Rocha, Daniel C. Hauagge, Jacques Wainer e Siome Goldenstein.

Este trabalho nos rendeu, inicialmente, o artigo [153] no *Brazilian Symposium of Computer Graphics and Image Processing* (Sibgrapi).

Chapter 5

Automatic Fruit and Vegetable Classification from Images

Abstract

In this paper, we address a multi-class fruit-and-vegetable categorization task in a semi-controlled environment, such as a distribution center or the supermarket cashier. To solve such a complex problem using just one feature descriptor is a difficult task and feature fusion becomes mandatory. Although normal feature fusion is quite effective for some problems, it can yield unexpected classification results when the different features are not properly normalized and prepared. Besides it has the drawback of increasing the dimensionality which might require more training data. To cope with these problems, we propose a unified approach that can combine many *features* and *classifiers*, requires less training, and is more adequate to some problems than a naïve method, where all features are simply concatenated and fed independently to each classification algorithm. Besides that, the algorithm proposed is amenable to continuous learning, both when refining a learned model and also when adding new classes to be discriminated. We validate the system using an image data set we collected at a local produce distribution center. Since this data set can help further research by the overall scientific community, we also make it publicly available over the internet.

5.1 Introduction

Recognizing different kinds of vegetables and fruits is a recurrent task in supermarkets, where the cashier must be able to point out not only the species of a particular fruit (i.e., banana, apple, pear) but also it's variety (i.e., Golden Delicious, Jonagold, Fuji), which will determine it's price. The use of barcodes has mostly ended this problem for packaged products but given that consumers want to pick their produce, they can not be packaged, and thus must be weighted. A common solution to this problem is issuing codes for each kind of fruit/vegetable; which has problems given that the memorization is hard, leading to errors in pricing.

As an aid to the cashier, many supermarkets issue a small book with pictures and codes; the problem with this solution is that flipping over the booklet is time-consuming. In this paper, we review several image descriptors in order to propose a system to solve the problem by adapting a camera to the supermarket scale that identifies fruits and vegetables based on color, texture, and appearance cues.

Formally, we state our problem in the following manner: given an image of fruits or vegetables of only one variety, in arbitrary position and number, the system must return a list of possible candidates of the form (species, variety). Sometimes, the object can be inside a plastic bag that can add specular reflections and hue shifts.

Given that the big variety and the impossibility of predicting which kinds of fruit/vegetables are sold, training must be done on site by someone with little or no technical knowledge. Therefore, the system must be able to achieve a high level of precision with only a few training examples (e.g., up to 30 images). Another desirable characteristic would be continuous learning. On one hand, more training data would be generated as the system commits mistakes and the cashier corrects them. On the other hand, in this semi-supervised scenario, eventually the operator will commit mistakes and the learning algorithm must be robust to noisy training data.

Here, we combine local and global features using different classification procedures. We used global color histograms, local texture, shape, and correlation descriptors with distinctive fruit parts.

Our contribution in this paper is twofold. The first is that we evaluate several image descriptors in the literature and point out the best ones to solve our multi-class fruits/vegetables categorization problem. Important questions about such descriptors are: which ones require less training? Is it necessary to use complex approaches such as bag-of-features or constellation models? How do the descriptors perform when increasing the number of training examples? What combinations and parameters of the descriptors provide better effectiveness? How do the descriptors behave under the curse-of-dimensionality? In the experiments we show answers for such questions.

We end up with a unified approach that can combine many features and classifiers that requires less training and is more adequate to some problems than a naïve method, where all features are simply concatenated and fed independently to each classification algorithm. Besides that, the algorithm proposed is amenable to continuous learning, both when refining a learned model and also when adding new classes to be discriminated.

The second contribution is that we create an image data set collected from our local fruits and vegetables distribution center and make it public. In general, there are a few well-documented image data sets freely available for testing algorithm performance in image categorization and content-based image retrieval tasks. In this context, we provide an image data set with 15 produce categories comprising 2,633 images collected on site with all its creation details. The images were collected in a period of five months under diverse conditions.

In Section 5.2, we give a brief overview of previous work in object recognition and image categorization. In Section 5.3, we present the different kinds of image descriptors we used in this paper as well as the produce data set. Section 5.4 presents results for diverse background subtraction approaches and analyzes the one we used in this paper. Section 5.5, reports results for the image data set we created for each feature and classifier with no fusion. Section 5.6 introduces our solution for feature and classifier fusion, and Section 5.7 presents experimental results. Section 5.8 draws some considerations about the proposed technique, as well as conclusions and future directions.

5.2 Literature review

Recently, there has been a lot of activity in the area of *Image Categorization*. Previous approaches considered patterns in color, edge and texture properties [126, 169, 178]; low- and middle-level features to distinguish broad classes of images [34, 98, 148, 162]; In addition, Heidemann [66] has presented an approach to establish image categories automatically using histograms, colors and shape descriptors with an unsupervised learning method.

With respect to our problem, *VeggieVision* [18] was the first attempt of a supermarket produce recognition system. The system uses color, texture and density (thus requiring extra information from the scale). However, as this system was presented long time ago, it does not take advantage of recent developments. The reported accuracy was ≈ 95% in some scenarios but to achieve such result it uses the top four responses. Our data set is also more demanding in some respects; while theirs had more classes the image capturing hardware gave a more uniform color and suppressed specular lights. The dataset we assembled have much greater illumination and color variation among images, also we take no measure to suppress specularities.

In general, we can view our problem as a special instance of object's categorization. Turk and Pentland [177] employed principal component analysis and measured the reconstruction error of projecting the image to a subspace and returning to the original image space. We believe this is ill suited for our purpose because it depends heavily on illumination, pose and shape.

Viola and Jones [182] presented an approach with localization speed and precision in recognition employing a cascade of classifiers composed of simple features and trained with the Ada-Boost algorithm. The drawback of this method is that its training is very costly, often requiring thousands of images.

Recently, Agarwal et al. [2] and Jurie and Triggs [75] adopted approaches that break down the categorization problem to the recognition of specific parts that are characteristic of each object class. These techniques, generally called bag-of-features [62, 68, 100], showed promising results even though they do not try to model spatial constraints among features.

Weber [189] takes into account spatial constraints using a generative constellation model. The algorithm can cope with occlusion in a very elegant manner, albeit very costly (exponential in the number of parts). A further development made by Fei-Fei et al. [46] introduced prior knowledge into the estimation of the distribution, thus reducing the number of training examples to around 10 images while preserving a good recognition rate. Notwithstanding, even with this improvement, the problem of exponential growth with the number of parts persists, which makes it unpractical for our problem, which requires speed for on-line operation.

Another interesting technique was proposed by Malik [15]. In this work, feature points are found in a gradient image. The points are connected by a joining path and a match is signalized if the found contour is similar enough to the one in the database. A serious drawback of this method for our problem is that it requires a nonlinear optimization step to find the best contour; besides that it relies too heavily on the silhouette cues, which are not a very informative feature for fruits like oranges, lemons and melons.

5.3 Materials and methods

In general, image categorization relies on combinations of statistical, structural and spectral approaches. In statistical approaches, we describe the objects using global and local descriptors such as mean, variance, and entropy. In structural approaches, we represent the object's appearance using well-known primitives such as patches of important parts of the object. Finally, in spectral approaches, we describe the objects using some spectral space representation such as Fourier spectrum [60]. In this paper, we analyze statistical color and texture descriptors as well as structural appearance descriptors to categorize fruits and vegetables in a multi-class scenario.

As the best combination of features was not known for our problem, we analyze several state-of-the-art computer vision features in many different ways, and assemble a system with good overall accuracy using underpinned cross-validation procedures that allows us to combine the best features and classifiers in a single and unified approach. Feature description combination is of particular interest in the literature and has demonstrated important results over the last years [88, 185]. However, the approach presented in [88] employs EXIF information which are not relevant for our produce classification system. In addition, the solution presented in [185] uses relevance feedback which would be a tiresome requirement to the supermarket cashier.

In the following, we present the statistical and structural descriptors we analyzed and used in this paper, as well as the data set we created for the validation process.

5.3.1 Supermarket Produce data set

The *Supermarket Produce* data set is one of our contributions in this paper[1]. In general, there are a few well-documented image data sets available for image categorization and content-based image retrieval tasks for testing algorithm performance. ALOI[2] and Caltech[3] are two examples of such data sets for general categorization. In this paper, we provide an image data set with 15 produce categories comprising 2,633 images collected on site.

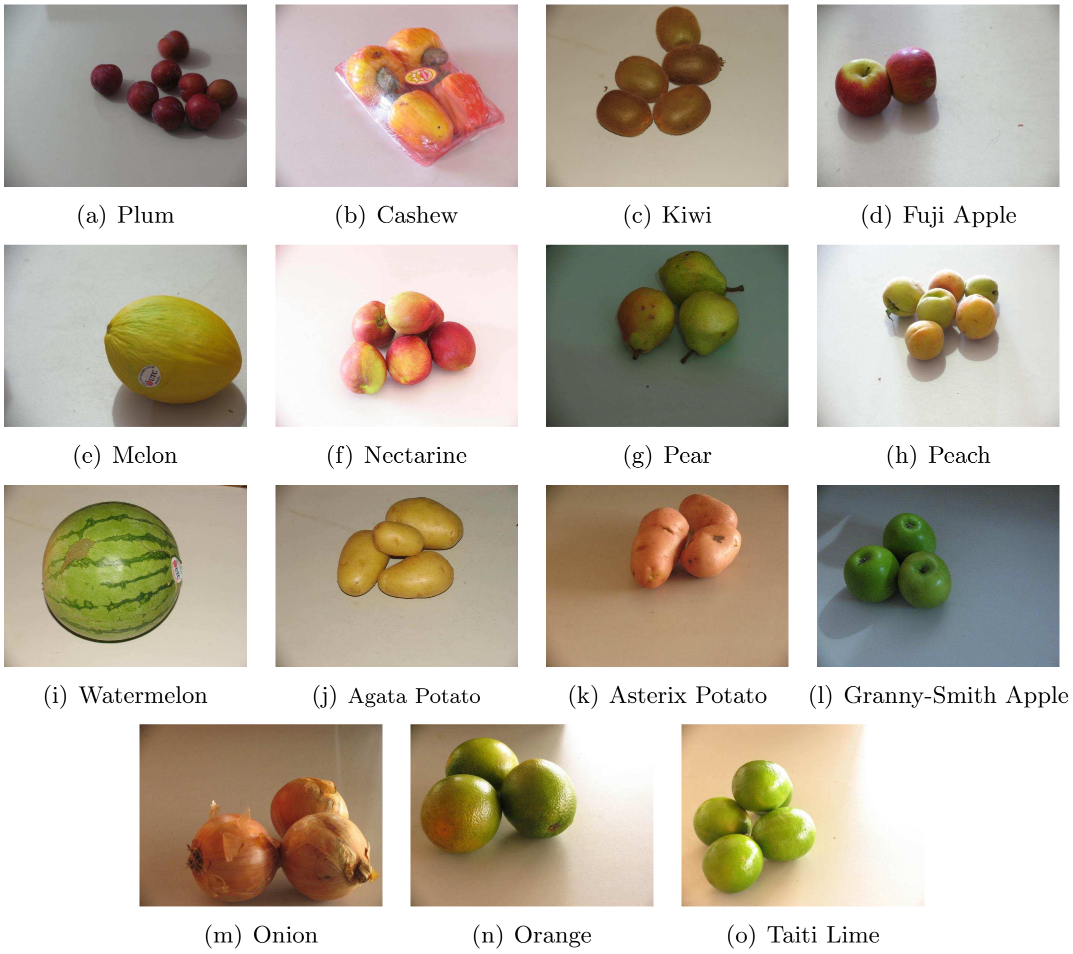
[1] Freely available from `http://www.liv.ic.unicamp.br/~undersun/pub/communications.html`

[2] `http://staff.science.uva.nl/~aloi`

[3] `http://www.vision.caltech.edu/Image_Datasets/`

The *Supermarket Produce* data set is the result of five months of on site collecting in our local fruits and vegetables distribution center.

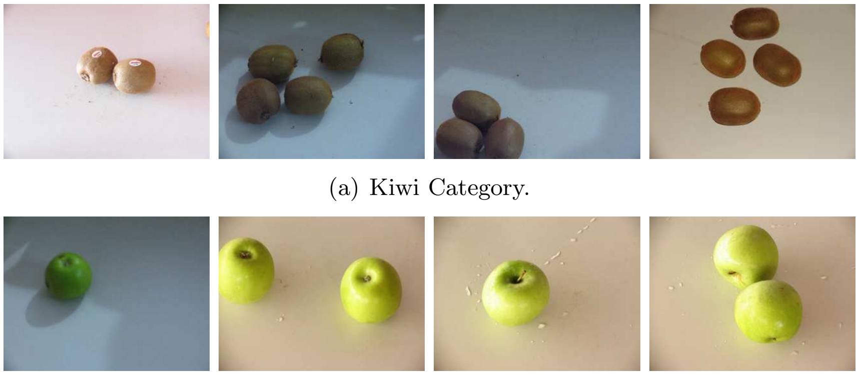
We used a Canon PowerShot P1 camera, at a resolution of $1,024 \times 768$ pixels. For the experiments in this paper we down-sampled the images to 640×480. For all images, we used a clear background. Illumination varies greatly among images. We acquired images from 15 different categories: Plum (264), Agata Potato (201), Asterix Potato (182), Cashew (210), Onion (75), Orange (103), Taiti Lime (106), Kiwi (171), Fuji Apple (212), Granny-Smith Apple (155), Watermelon (192), Honeydew Melon (145), Nectarine (247), Williams Pear (159), and Diamond Peach (211); totalizing 2,633 images. Figure 5.1 depicts some of the classes of our image data set.



(a) Plum (b) Cashew (c) Kiwi (d) Fuji Apple (e) Melon (f) Nectarine (g) Pear (h) Peach (i) Watermelon (j) Agata Potato (k) Asterix Potato (l) Granny-Smith Apple (m) Onion (n) Orange (o) Taiti Lime

Figure 5.1: *Supermarket Produce* data set.

All the images were stored in RGB color-space at 8 bits per channel. We gathered images at various times of the day and in different days for the same category. These features increase the data set variability and represent a more realistic scenario. Figure 5.2 shows an example of Kiwi and Granny-Smith Apple categories with different lighting. The differences are due to illumination, no image pre-processing was done.



(a) Kiwi Category.

(b) Granny-Smith Apple Category.

Figure 5.2: Illumination differences within categories.

The *Supermarket Produce* data set also comprises differences in pose and in the number of elements within an image. Figure 5.3 shows examples of the Cashew category. Note that there are variations in the pose of the Cashew's plastic repository. In addition, Figure 5.4 shows the variability in the number of elements within an image.

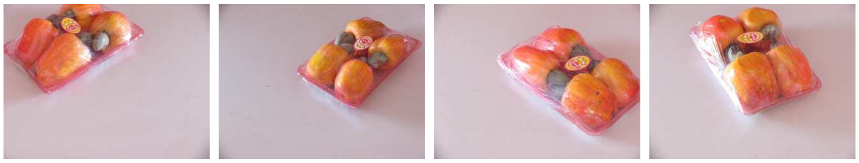


Figure 5.3: Pose differences. Cashew category.

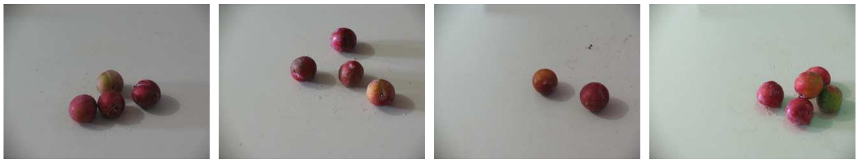


Figure 5.4: Variability on the number of elements. Plum category.

Sometimes, the elements are inside a plastic bag which adds specular reflections to the analyzed image. Furthermore, the presence of shadows (e.g., second and third images of Figure 5.2(a)) and cropping/occlusions (e.g., Figure 5.5) makes the data set more realistic.

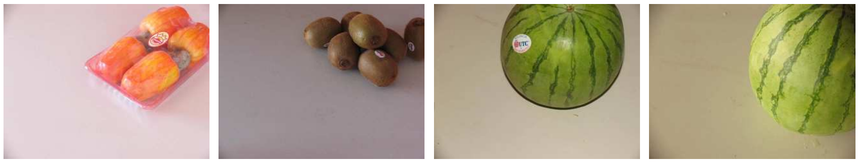


Figure 5.5: Examples of cropping and partial occlusion.

5.3.2 Image descriptors

In this section, we analyze statistical color, texture, and structural appearance descriptors (bag-of-features) in order to propose a system to solve a multi-class fruits/vegetables categorization problem.

Global Color Histogram (GCH)

The simplest approach to encode the information present in an image is the Global Color Histogram (GCH) [60]. A GCH is a set of ordered values, one for each distinct color, representing the probability of a pixel being of that color. Uniform quantization and normalization are used to reduce the number of distinct colors and to avoid scaling bias [60]. In this paper, we use a 64-d GCH feature vector.

Unser's descriptors

Unser [178] has shown that the sum and difference of two random variables with same variances are de-correlated and define the principal axes of their associated joint probability function. Hence, the author introduces sum s and difference d histograms as an alternative to the usual co-occurrence matrices for image texture description.

The non-normalized sum and difference associated with a relative displacement (δ_1, δ_2 for an image I, are defined as

$$s_{k,l} = I_{k,l} + I_{k+\delta_1,l+\delta_2}, \tag{5.1}$$

$$d_{k,l} = I_{k,l} - I_{k+\delta_1,l+\delta_2}. \tag{5.2}$$

The sum and difference histograms over the domain D are defined in a manner similar to the spatial level co-occurrence or dependence matrix definition:

$$\begin{aligned} h_s(i;\delta_1,\delta_2) = h_s(i) &= Card\{(k,l) \in D, s_{k,l} = i\}, && (5.3)\\ h_d(j;\delta_1,\delta_2) = h_d(j) &= Card\{(k,l) \in D, d_{k,l} = j\}. && (5.4) \end{aligned}$$

In addition to the histograms, as we show in Table 5.1, we use some associated global measures: mean (μ), contrast (C_n), homogeneity (H_g), energy (E_n), variance (σ^2), correlation (C_r), and entropy (H_n)) over the histograms.

Mean	$\mu = \frac{1}{2}\sum_i i h_s[i]$
Contrast	$C_n = \sum_j j^2 h_d[j]$
Homogeneity	$H_g = \frac{1}{1+j^2} h_d[j]$
Energy	$E_n = \sum_i h_s[i]^2 \sum_j h_d[j]^2$
Variance	$\sigma^2 = \frac{1}{2}\left(\sum_i (i-2\mu)^2 h_s[i] + \sum_j j^2 h_d[j]\right)$
Correlation	$C_r = \frac{1}{2}\left(\sum_i (i-2\mu)^2 h_s[i] - \sum_j j^2 h_d[j]\right)$
Entropy	$H_n = -\sum_i h_s[i] \log\left(h_s[i]\right) - \sum_j h_d[j] \log\left(h_d[j]\right)$

Table 5.1: Histogram-associated global measures.

In this paper, we use a 32-d Unser feature vector calculated in the grayscale representation of the images.

Color Coherence Vectors (CCVs)

Pass et al. [126] presented an approach to compare images based on color coherence vectors. They define color's coherence as the degree to which pixels of that color are members of a large region with homogeneous color. They refer to these significant regions as coherent regions. Coherent pixels are part of some sizable contiguous region, while incoherent pixels are not.

In order to compute the CCVs, first the method blurs and discretizes the image's color-space to eliminate small variations between neighboring pixels. Thereafter, it finds the connected components in the image aiming to classify the pixels within a given color bucket as either coherent or incoherent.

After classifying the image pixels, CCV computes two color histograms: one for coherent pixels and another for incoherent pixels. The two histograms are stored as a single histogram. In this paper, we use a 64-d CCV feature vector.

Border/Interior (BIC)

Stehling et al. [169] presented the border/interior pixel classification (BIC), a compact approach to describe images. BIC relies on the RGB color-space uniformly quantized in $4 \times 4 \times 4 = 64$ colors. After the quantization, the image pixels are classified as *border* or *interior*. A pixel is classified as *interior* if its 4-neighbors (top, bottom, left, and right) have the same quantized color. Otherwise, it is classified as *border*.

After the image pixels are classified, two color histograms are computed: one for border pixels and another for interior pixels. The two histograms are stored as a single histogram with 128 bins.

Appearance descriptors

To describe local appearance, we use a vocabulary of parts, similar to Agarwal et al. [2] and Jurie and Triggs [75]. Images are converted to grayscale to locate interest points and patches are extracted from the gradient magnitude image or the original grayscale image. We use Lowe's feature point detector to find the coordinates of interest points, together with orientation and scale. Once found, we extract a square region around the point. The square side is proportional to the scale and the orientation follows that of the feature point. Once extracted, all patches are resized to 13 × 13 pixels.

All patches in the training set are clustered using K-means. The cluster centers are used as our part dictionary. The found centroids can be seen on Figure 5.6.

To compute the feature vector for a given image, we extract patches in the same way that was done for the training set. The feature vector length is equal to the number of parts in our dictionary. We tried two schemes for values of the components of the feature vectors. In the first one the value for each component is equal to the distance between the dictionary part d_i and the closest patch p_j in the given image, as in the equation

$$f_i = \min_{\forall j} \frac{p_j \cdot d_i}{\|p_j\| \|d_i\|}. \tag{5.5}$$

In the second scheme, the value for the component is equal to 1 if this part is the closest one for some patch of the input image, and 0 otherwise.

$$f_i = \begin{cases} 1 & \text{if } i = \arg\min\limits_i \frac{p_j \cdot d_i}{\|p_j\| \|d_i\|} \text{ for some } j \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

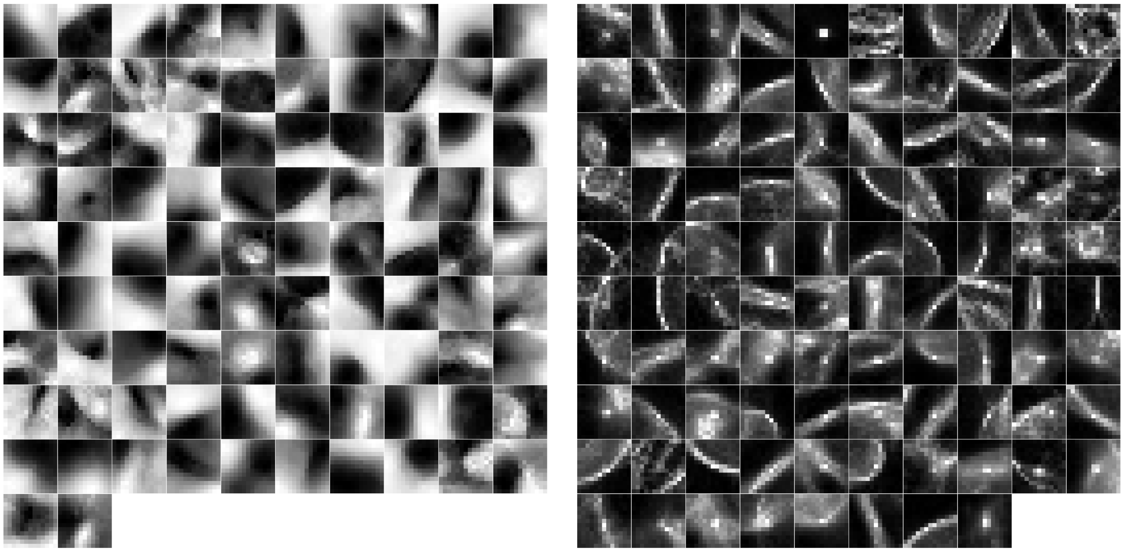
When convenient, we show the name of the algorithm used and the size of the used feature vector. For instance, K-Means-98 refers to the use of K-Means algorithm on a code-book (feature space) of 98 dimensions.

For the vocabulary of parts, we use some images from the *Supermarket Produce* data set in the vocabulary creation stage. The images used for the vocabulary generation are excluded from the data set in the posterior training/testing tasks.

5.3.3 Supervised learning

Supervised learning is a machine learning approach that aims to estimate a classification function f from a *training data set*. Such training data set consists of pairs of input values X and its desired outcomes Y [55]. Observed values in X are denoted by x_i, i.e., x_i is the i^{th} observation in X. Often, x is as simple as a sequence of numbers that represent some observed features. The number of variables or features in each $x \in X$ is p. Therefore, X is formed by N input examples (vectors) and each input example is composed by p features or variables.

The commonest output of the function f is a label (class indicator) of the input object under analysis. The learning task is to predict the function outcome of any valid input object after



(a) Grayscale.

(b) Gradients.

Figure 5.6: Dictionary of parts, clustered using K-means.

having seen a sufficient number of training examples.

In the literature, there are many different approaches for supervised learning such as Linear Discriminant Analysis, Support Vector Machines, Classification Trees, and Neural Networks.

Linear Discriminant Analysis

Also known as Fisher discriminant, Linear Discriminant Analysis (LDA) consists of selecting the components that maximize the between-class differences while minimizing the within-class variability [16].

For the sake of simplicity, consider a two-class classification problem Y(-1 = a | +1 = b). Let X_a be a set of examples belonging to the class a and X_b be a set of examples belonging to class b in the training set. Consider the number of elements in X_a to be N_a and the number of elements in X_b to be N_b.

Let's suppose that both classes a and b have a Gaussian distribution. Therefore, we can define the within-class means as

$$\mu_a = \frac{1}{N_a} \sum_{x_i \in X_a} x_i \qquad \text{and} \qquad \mu_b = \frac{1}{N_b} \sum_{x_j \in X_b} x_j. \tag{5.7}$$

We define the between-class (a, b) means as

$$\mu = \frac{1}{N_a + N_b} \left(\sum_{x \in X_a \cup X_b} x \right). \tag{5.8}$$

We define the within-class scatter matrix S_w as

$$S_w = M_a M_a^T + M_b M_b^T, \tag{5.9}$$

where the i^{th} column of matrix M_a contains the difference $(x_i^a - \mu_a)$. We can apply the same procedure to M_b. The scatter matrix S_{bet} between the classes is

$$S_{bet} = N_a(\mu_a - \mu)(\mu_a - \mu)^T + N_b(\mu_b - \mu)(\mu_b - \mu)^T. \tag{5.10}$$

In order to maximize the between-class differences and to minimize the within-class variability in one single dimension (1-D), it is enough to calculate the generalized eigenvalue-eigenvector $\vec{e}$ of S_{bet} and S_w

$$S_{bet}\vec{e} = \lambda S_w \vec{e}.$$

With the generalized eigenvalue-eigenvector, we can project the samples to a linear subspace and use a single threshold to perform the classification.

Support Vector Machine (SVM)

In the SVM model, we are looking for an optimal separating hyper-plane between two classes. This is done maximizing the *margin* (minimal distance of an example to the decision surface). When it is not possible to find a linear separator, we project the data into a higher space using techniques known as kernels [16]. SVMs can be linear separable, linear non-separable and non-linear.

In the linear separable SVMs, the points that lie on the hyper-plane satisfy the constraint

$$w^t x_i + b = 0, \tag{5.11}$$

where w is the normal to the hyper-plane, b is the bias of the hyper-plane, $|b|/\|w\|$ is the perpendicular distance from the origin to the hyper-plane and $\|.\|$ is the Euclidean norm. To find out w and maximize the margin, we can use Lagrangian multipliers

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j x_i^t x_j y_i y_j, \tag{5.12}$$

where α is the positive Lagrangian multipliers, x are the samples of the X input vector and y are the expected outputs. A solution to the linear separable classifier, if it exists, yields values of α_i. Using the α_i values, we can calculate the normal to the hyperplane w

$$w = \sum_{i=1}^{N} \alpha_i x_i y_i. \tag{5.13}$$

From w we can calculate the bias b of the separating hyperplane applying the Karush-Kuhn-Tucker [16] condition

$$b = \frac{1}{N} \sum_{i=1}^{N} y_i - w^t x_i. \tag{5.14}$$

Sometimes, a linear separable SVM may not find a solution. Such a situation can be handled introducing *slack* variables ϵ_i that relax the constraint in the equation 5.11 [16].

When it is not possible to find a linear hyperplane that optimally separates the data, we can apply a non-linear SVM. In this model, we map the training examples into a higher-dimensional Euclidean space in which a linear SVM can be applied. This mapping can be done using a Kernel K such a polynomial or a radial basis function (RBF).

Denoting by $\phi = \mathcal{L} \rightarrow \mathcal{H}$, a mapping from the lower to the higher space and $K(x_i, x_j) = \phi(x_i)^t \phi(x_j)$ an on-the-fly kernel, we can test a new incoming exemplar z simply solving the equation

$$w^t \phi(z) + b = \sum_{i=1}^{N} (\alpha_i K(x_i, z) y_i + b). \tag{5.15}$$

Bootstrap Aggregation

In Bagging (Bootstrap aggregation) ensemble, we repeatedly apply an inductive learner (classifier) to bootstrap samples on the training set. We use the training set to generate bootstrap samples using random sampling with replacement. Once several hypotheses (i.e., base learners) have been generated on such bootstraps, we determine the aggregate classifier by majority voting among the base learners. The final classifier evaluates test samples by querying each of the base classifiers on the sample and then outputting their majority opinion.

Let X be our input data set, and Z^i, $i = 1, 2, \ldots, B$, one sample of X. To perform the classification in each Z^i, we select a weaker classifier (e.g., LDA). Often, we use the same classifier in all Z^i samples[4].

Figure 5.7 depicts the training and classification approaches using Bagging ensemble.

We store the coefficients related to each weak classifier used (α) such that when we analyze an unseen input example, we submit this example to the B classifiers and perform the final voting to predict the input example's class.

Clustering

Sometimes, we do not have a complete information about the data set under analysis (e.g., the class of all elements). One way to solve this problem consists of partitioning the data set into subsets (clusters), so that the data in each subset (ideally) share some common characteristics. The computational task of classifying a data set into k clusters is often referred to as k-clustering [16, 55].

The simplest approach to cluster data into similar groups is K-Means [16]. In this procedure, we define k centroids, one for each cluster. The better choice is to place the centroids far away from each other. Next, we take each point belonging to a given data set and associate it to the closest centroid. After analyzing all data points, we complete the first step and we have a set of k clusters. At this point, we re-calculate k new centroids as the barycenters of the resulting

[4] The number of times that we repeat the process is referred to as the number of bagging iterations.

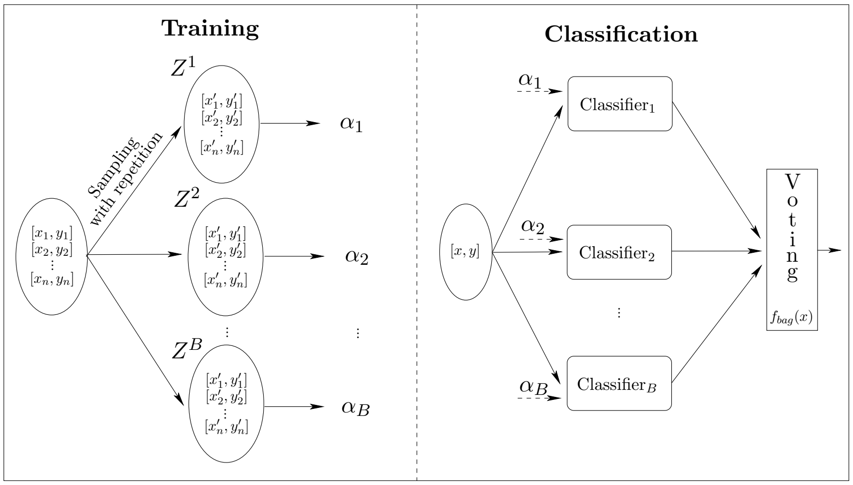


Figure 5.7: Training and classification using Bagging ensemble.

clusters from the previous step. With the k new centroids, we perform a new binding between the same data set points and the closest new centroid. We repeat this process until we reach a stabilization criterion.

5.4 Background subtraction

Background subtraction is a convenient and effective method for detecting foreground objects in images with stationary background.

This task has been widely studied in the literature. Background subtraction techniques can be seen as a two-object image segmentation and, often, need to cope with illumination variations and sensor capturing artifacts such as blur. For our problem we also face specular reflections, background clutter, shading and shadows.

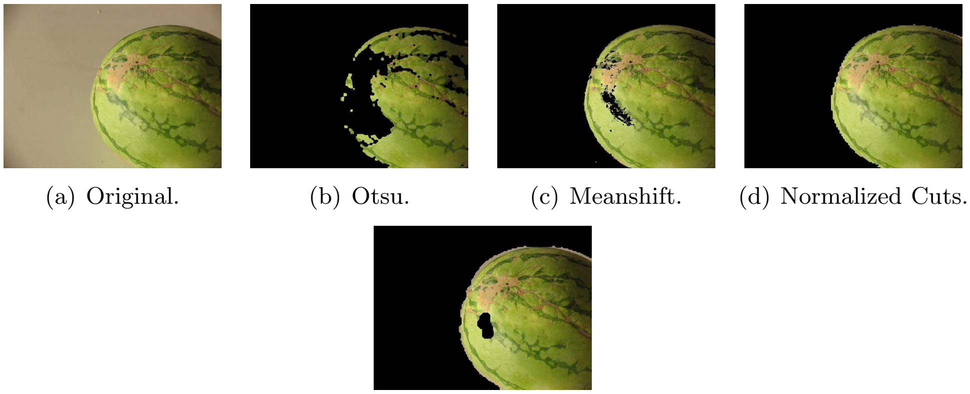
For a real application in a supermarket, background subtraction needs to be fast, requiring only fractions of a second to be performed.

In this paper, we tested several background subtraction techniques. After analyzing the cost effectiveness of each one, we present a solution that gives us good solutions in less than a second. In the following, we present some results for the different approaches we studied.

For the following experiments, we used a 2.1GHz machine with 2GB of RAM. No other program was executed in parallel. We discovered that, for our problem, the best channel to perform the background subtraction is the S channel of HSV-stored images. This is understandable given that the S channel is much less sensitive to lighting variations than any of the RGB color channels [60]. Therefore, for all tests reported, we performed the segmentation in the S channel.

Figure 5.8 depicts results for four different approaches. Otsu background algorithm [124] is the fastest tested approach requiring only 0.3 seconds to segment an input image of 640 × 480 pixels. However, as we see, it misses most of the produce under analysis. Meanshift [30] provides a good image segmentation within 1.5 seconds in average. However, it requires the correct tuning of parameters for different image classes. The affinity threshold for merging regions, also known as bandwidth in Meanshift, is different for each image class and sometimes even within the same class. Although, in general, the Normalized cuts [165] approach provided the best segmentation results, it has a serious drawback for our problem. Given that the approach needs to calculate the eigenvectors of the image, even for a reduced image (128×96 pixels), it requires ≈ 5 seconds to perform the segmentation, not counting the image resizing operation.

Finally, K-Means is the approach that yields good segmentation results within acceptable computing time. To analyze an image of 640 × 480, it requires 0.8 seconds for the whole procedure described in Algorithm 6. We also tested Watershed transform [90] followed by some morphological operations and the results are similar to Otsu's algorithm.



(a) Original. (b) Otsu. (c) Meanshift. (d) Normalized Cuts.

(e) K-Means.

Figure 5.8: Background subtraction results for four different approaches.

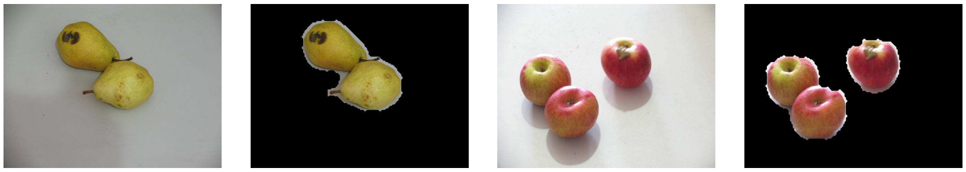
In this sense, we chose the approach based on K-Means to perform the background subtraction and reduce most of the artifacts due to lighting variation conditions. Algorithm 6 presents the complete procedure for this task.

Algorithm 6 Background subtraction based on K-Means

```
Require: Input image $I$ stored in HSV;
$I_{down} \leftarrow$ Down-sample the image to 25% of its original size using simple linear interpolation.
Get the $S$ channel of $I_{down}$ and consider it as an 1-d vector $V$ of pixel intensities.
Perform $D_{bin}$ = K-Means($V$, k=2)                                        ▷ Sec. 5.3.3
Map $M \leftarrow D_{bin}$ back to image space. For that just do a linear scan of $D_{bin}$.
$M_{up} \leftarrow$ Up-sample the generated binary map $M$ back to the input image size.
Close small holes on $M_{up}$ using the Closing morphological operator with a disk structuring element
    of radius 7 pixels.
```

With the generated binary map, in the stage of feature extraction, we limit all the features to be calculated within the object region of the masks. Figure 5.9 depicts some more background subtraction results.

It is worth noting that all descriptors are calculated within the segmented object's mask discarding the boundaries information. In other words, we do not calculate a descriptor info on the borders. In addition, it is hard to define a golden standard measure for the effectiveness of the segmenation regardless the selected procedure. However, we can point out that the segmentation quality using K-Means on HSV-represented images most likely is above 95% quality for the tested data set.



(a) Original. (b) K-Means. (c) Original. (d) K-Means.

Figure 5.9: Some K-Means background subtraction results.

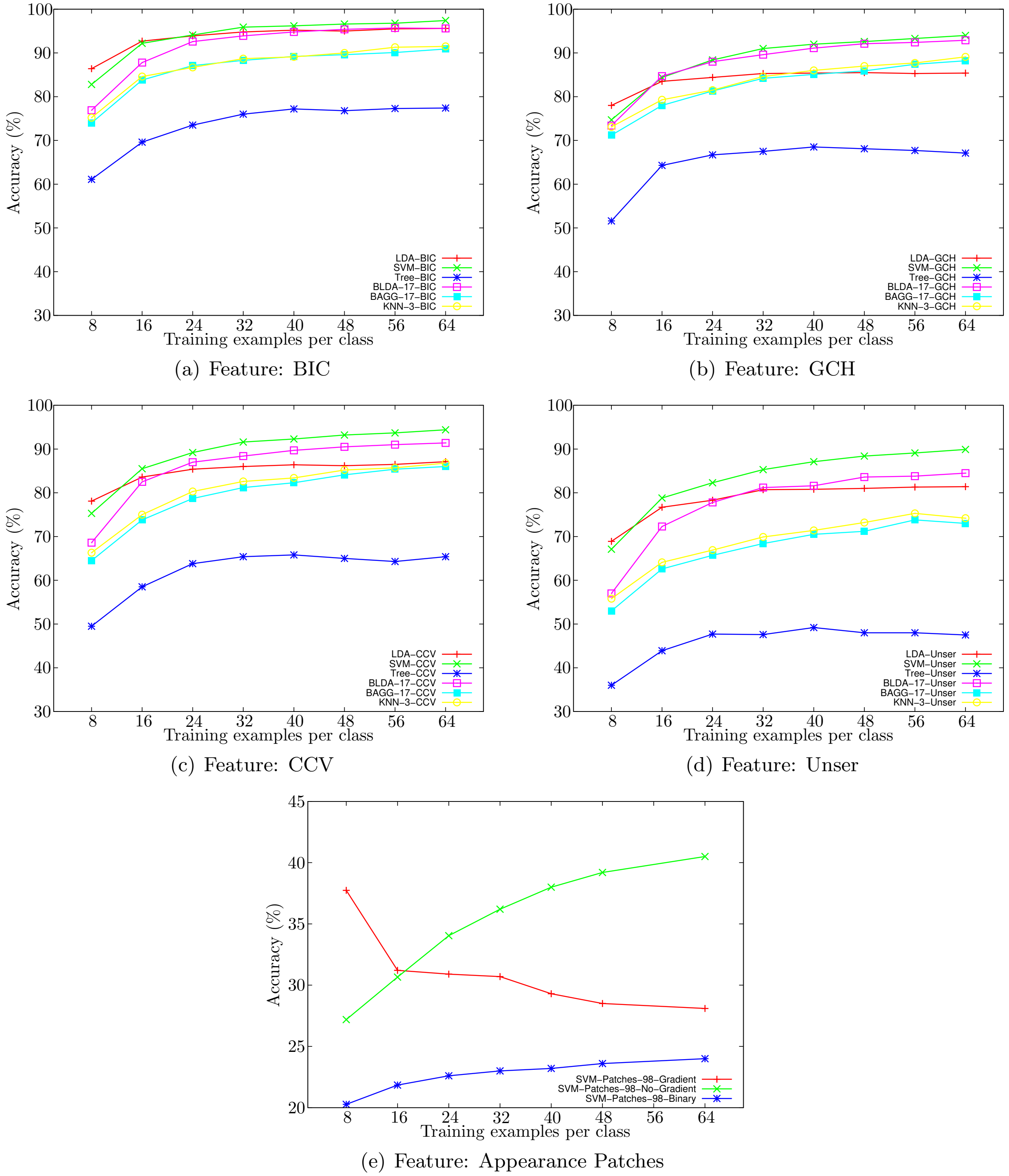
5.5 Preliminary results

In this section, we present preliminary results for our problem. In the quest for finding the best classification procedures and features, first we analyze several appearance-, color-, texture-, and shape-based image descriptors as well as diverse machine learning techniques such as SVM, LDA, Trees, K-NN, and Ensembles of Trees and LDA [16].

We select the training images using sampling without replacement from the pool of each image class. If we are training with 10 images per class, we use the remaining ones for testing. We repeat this procedure 10 times, and report the average classification accuracy (μ), average error ($\epsilon = 1 - \mu$), and standard deviation (σ). We do not use the strictly 10-fold cross-validation, given that we are interested in different sizes of training sets. In each round, we report the accuracy for the 15 classes summing the accuracy of each class and dividing by the number of classes.

Average accuracy rates

In Figure 5.10, we show results for different combinations of features and classifiers. The x-axis represents the number of images per class in the training set and the y-axis represents the average accuracy in the testing set.



(a) Feature: BIC

(b) Feature: GCH

(c) Feature: CCV

(d) Feature: Unser

(e) Feature: Appearance Patches

Figure 5.10: Average accuracy per class considering diverse classifiers and features.

In this experiment, we see that Breiman's decision Tree does not perform very well. One possible explanation is that the descriptor data is not suitable for this kind of classifier. For instance, the Unser descriptor feature vector is continuous and unbounded which naturally makes decision trees unstable.

In this sense, the ensemble of trees (BAGG), with 17 iterations performs better than simple decision Trees as far as it is more stable.

We also observe across the plots that LDA accuracy curves practically become flat for more than 32 examples. When we use an ensemble of LDA (BLDA), we are performing random sampling across the training data that makes a better use of the provided information in such a way it can improve the classification. Therefore, as we observe in the plots, the ensemble of LDA with 17 iterations performs better than straight LDA, Trees, or ensemble of Trees. Complementing, the simple K-Nearest Neighbors here is as good as ensemble of Trees no matter the descriptor. K-NN is also not well suited for unbounded and unrestricted data like Unser descriptor (c.f., Fig. 5.10(d)). In particular, SVM is a classifier that performs well regardless the used features.

As we can see, for this problem, BIC descriptor performs best yielding an accuracy $\approx 94\%$ for 32 examples per class in the training under SVM classifier.

Furthermore, a more complex approach such as the appearance descriptor for this particular problem does not yield a good classification accuracy, as Figure 5.10(e) depicts. We tested three different approaches for the appearance-based descriptor in Figure 5.10(e). Two interesting observations: (1) the approach based on patches with no gradient orientation is the only tested feature which does not benefit from more examples in the training; and (2) the approach based on patches with gradient orientation is the one which benefits more with more training examples. This suggests us that with enough training it might provide much better results. Notwithstanding, the training data is limited for the particular problem in this paper.

We believe that the appearance descriptor does not provide a significant classification accuracy given that the used patches do not represent well all the images classes we are trying to classify. Further investigation must be done in this direction to validate the use of appearance descriptor or any other similar model. However, one observation that pops up is that this model requires more sophisticated training. Although, previous results in literature argue that it requires less examples for training, we noted that it requires lots of good representative images to create good appearance patches and accomplish such claims.

We can draw some important general conclusions when analyzing these results. First, it is hard to solve a multi-class problem using just one feature descriptor. Hence feature fusion becomes mandatory. Second, we see that the classifier results for different features are quite different suggesting us that it would be worth also combining the classifiers each one tailored and best-suited for a particular feature.

In spite of the fact that feature-level combination is not straightforward for multi-class problems, for binary problems this is a simple task. In this scenario, it is possible to combine different classifiers and features by using classic rules such as `and`, `or`, `max`, `sum`, or `min` [16]. For multi-class problems, this is more difficult given that one feature might point out to an outcome class

C_i and another feature might result the outcome class C_j, and even another one could result C_k. With many different resulting classes for the same input example, it becomes difficult to define a consistent policy to combine the selected features.

In this context, one approach sometimes used is to combine the feature vectors for different features into a single and big feature vector. Although quite effective for some problems, this approach can also yield unexpected classification results when not properly prepared. First, in order to create the combined feature vector, we need to tackle the different nature of each feature. Some can be well conditioned such as continuous and bounded variables, others can be ill-conditioned for this combination such as categorical ones. In addition, some variables can be continuous and unbounded. To put everything together, we need a well-suited normalization. However, this normalization is not always possible or, sometimes, it leads to undesirable properties in the new generated feature vector such as equally weighting all the feature coefficients, a property that in general we do not want.

When combining feature vectors this way, eventually we would need to cope with the curse of dimensionality. Given that we add new features, we increase the number of dimensions which then might require more training data.

Finally, if we want to add a new feature, we might need to redesign the normalization in order to deal with all the aforementioned problems. In Section 5.6, we present a simple and effective way to feature and classifier fusion that cope with most of the previously discussed concerns.

5.6 Feature and classifier fusion

Our objective is to combine a set of features and the most appropriate classifier for each one in such a way we improve the overall classification accuracy. Given that we do not want to face the inherent problems of proper normalization and curse of dimensionality, we do not create a big feature vector combining the selected features. Furthermore, doing that we would only perform feature fusion and we would still be limited in doing the classifier fusion.

To accomplish our objective, we propose to cope with the multi-class problem as a set of binary problems. In this context, we define a *class binarization* as a mapping of a multi-class problem onto two-class problems (divide-and-conquer) and the subsequent combination of their outcomes to derive the multi-class prediction. We refer to the binary classifiers as *base learners*. Class binarization has been used in the literature to extend naturally binary classifiers to multi-class and SVM is one example of this [5,38,115]. However, to our knowledge, this approach was not used before for classifier and feature fusion.

In order to understand the class binarization, consider a problem with 3 classes. In this case, a simple binarization consists in training three base learners, each one for two classes. In this sense, we need $O(N^2)$ binary classifiers, where N is the number of classes.

We train the ij^{th} binary classifier using the patterns of class i as positive and the patterns of class j as negative examples. To obtain the final outcome we just calculate the minimum distance of the generated vector (binary outcomes) to the binary pattern representing each class.

Consider again a toy example with three classes as we show in Figure 5.11. In this example, we have the classes: *Triangles* $\triangle$, *Circles* $\bigcirc$, and *Squares* $\square$. Clearly, one first feature we can use to categorize elements of these classes can be based on shape. As we can see, we can also use texture and color properties. To solve this problem, we train some binary classifiers differentiating two classes at a time, such as $\triangle \times \bigcirc$, $\triangle \times \square$, and $\bigcirc \times \square$. Also, we give each one of our classes a unique identifier (e.g., $\triangle = \langle +1, +1, 0 \rangle$).

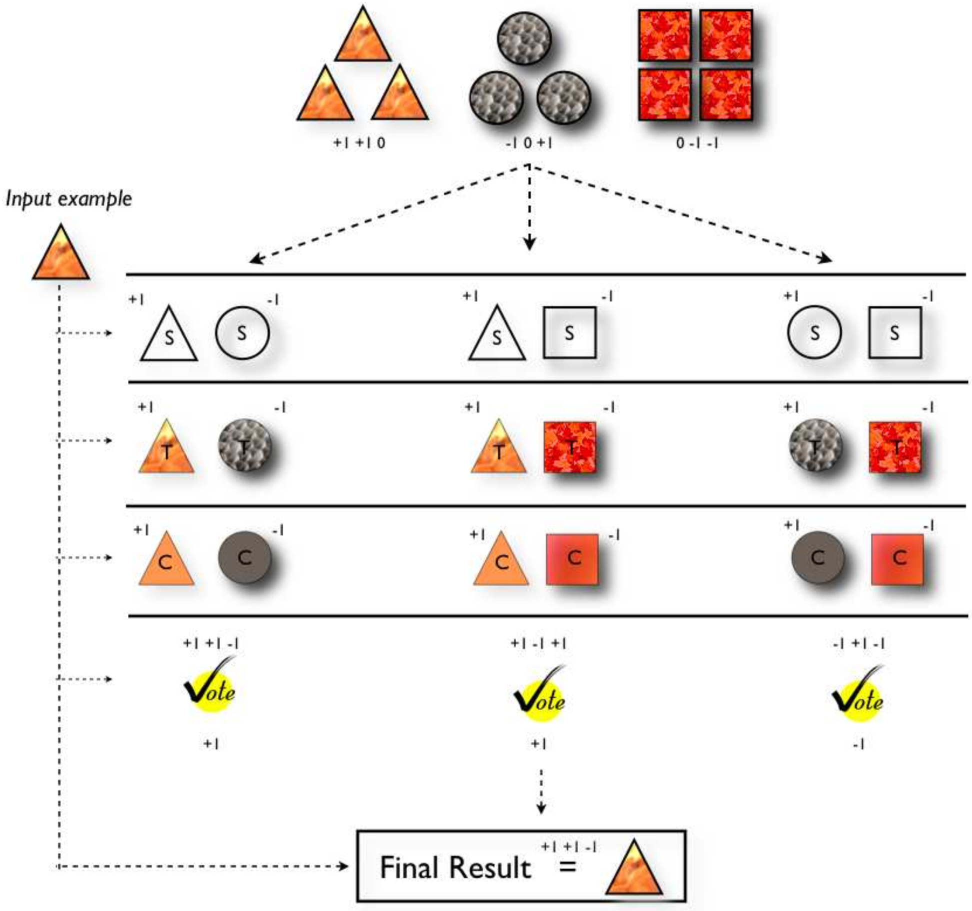


Figure 5.11: Toy example for feature and classifier fusion.

When we receive an input example to classify, let's say a triangle-shaped one, as we show in the picture, we first apply our binary classifiers to verify if the input example is a triangle or a circle based on shape, texture and color features. Each classifier will give us a binary response. Let's say we obtain the votes $\langle +1, +1, -1 \rangle$ for the binary classifier $\triangle \times \bigcirc$. Thereafter, we can use majority voting and select one response (+1 in this case, or $\triangle$). Then we repeat the procedure and test if the input example is a triangle or a square, again for each one of the considered features. Finally, after performing the last test, we end up with a binary vector. Then we calculate the minimum distance from this binary vector to each one of the class unique IDs. In this example, the final answer is given by the minimum distance of

$$\min dist(\langle 1, 1, -1\rangle, \{\langle 1, 1, 0\rangle, \langle -1, 0, 1\rangle, \langle 0, -1, -1\rangle\}. \tag{5.16}$$

One aspect of this approach is that it requires more storage given that once we train the binary classifiers we need to store their parameters. Given that we analyze more features we need more space. With respect to running time, there is also a small increase given that we need to test more binary classifiers to provide an outcome. However, many classifiers in our daily-basis employ some sort of class binarization (e.g., SVM) and are considered fast. The majority voting for each binary classifier and the distance calculation to the unique class IDs are simple and efficient operations.

Although we require more storage and increase the classification time with respect to a normal multi-class approach, there are some advantages using this approach:

1. By combining independent features, we have more confidence in a resulting outcome given that it is calculated from the agreement of more than one single feature. Hence, we have a simple error correcting mechanism that can withstand some misclassifications;

2. If we find some classes that are in confusion and are driving down our classification results, we can design special and well-tuned binary classifiers and features to separate them;

3. We can easily point out if one feature is indeed helping the classification or not. This is not straightforward with normal binding in a big feature vector;

4. The addition of new classes only require training for the new binary classifiers;

5. The addition of new features is simple and also just require partial training;

6. As we do not increase the size of any feature vector, we are less prone to the curse of dimensionality not requiring more training examples when adding more features.

Finally, there is no requirement to combine the binary classifiers using all combinations of two classes at a time. We can reduce storage requirements and speed up the classification itself by selecting classes that are in confusion and designing specific binary classifiers to separate them. The expectation in this case is that much less binary classifiers would be needed. In fact, there is room for more research in this direction.

5.7 Fusion results

We have seen in Section 5.5 that it is hard to solve a complex problem such as the one in this paper using just one feature descriptor. Therefore, we face the need for feature fusion to improve classification. In addition, we also observed that some classifiers are better for some features than others which suggests us that it would be worth also combining the classifiers, each one tailored and best-suited for a particular feature. We also discussed that common feature fusion approaches, although effective and powerful in some cases, require careful understanding of the used features and their proper normalization. Furthermore, this kind of feature combination increase the dimensionality of the feature vectors and may require more training examples. In this context, we presented in Section 5.6 an alternative approach that allows us to combine features and classifiers in a simple way.

In this section, we present results for our approach and show that the combined features and classifiers indeed improve classification results when compared to the standalone features and classifiers. In the following experiments, we report results showing the average error in classification and we seek to minimize this error. Finally, we show that with the top first responses we have less than 2% error in classification and with the top two we have less than 1%.

5.7.1 General fusion results

Figure 5.13 shows one example of the combination of the BIC, CCV, and Unser descriptors. This combination is interesting given that, BIC is a descriptor that analyzes color and shape in the sense that it codifies the object's border and interior, CCV codifies the color connected components and Unser accounts for the image's textures.

In Figure 5.13, we see that the fusion works well regardless the classifier used. Consider the SVM classifier, with 32 examples per class in the training, the fusion yields an average error of $\epsilon = 3,0\%$ and standard deviation of $\sigma = 0.43\%$. This is better than the best standalone feature, BIC, that is $\epsilon = 4.2\%$ and standard deviation of $\sigma = 0.32\%$. Although the absolute difference here seems small, it is about 3 standard deviations which means it is statistical significant. In general, to reduce one percentual point in the average error when the baseline accuracy is $\approx 95\%$ is a hard problem. For LDA classifier, the fusion requires at least 24 examples per class in the training to yield error reduction. Recall that we observed in Section 5.5 that LDA curves become flat with more than 32 examples per class in the training and adding more training data does yield better results. On the other hand, when combining different features, LDA does benefit from more training and indeed results lower error rates ($\epsilon = 3\%, \sigma = 0.59\%$), 9.8 standard deviations better than LDA on the straight BIC feature.

In Figure 5.14, we switch the BIC descriptor to a simpler one with half of the dimensionality (64-d). As we note, the results are comparable to the ones obtained with the fusion before. But now, the fusion show even more power.

5.7.2 Top two responses

Figure 5.12 shows the results when we require the system to show the top 2 responses. In this case, the system provides the user the two most probable classes for a given input example considering the different classifiers and features used. Using SVM classifier and fusion of BIC, GCH, and Unser features, with 32 examples per class in the training, the average error is $\epsilon \leq 1\%$.

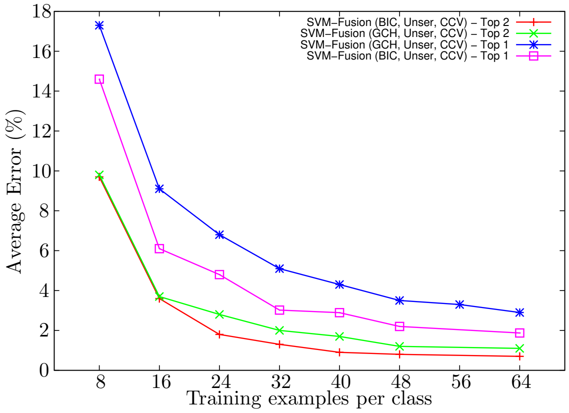


Figure 5.12: Top one, and two responses for SVM classifier considering fusion.

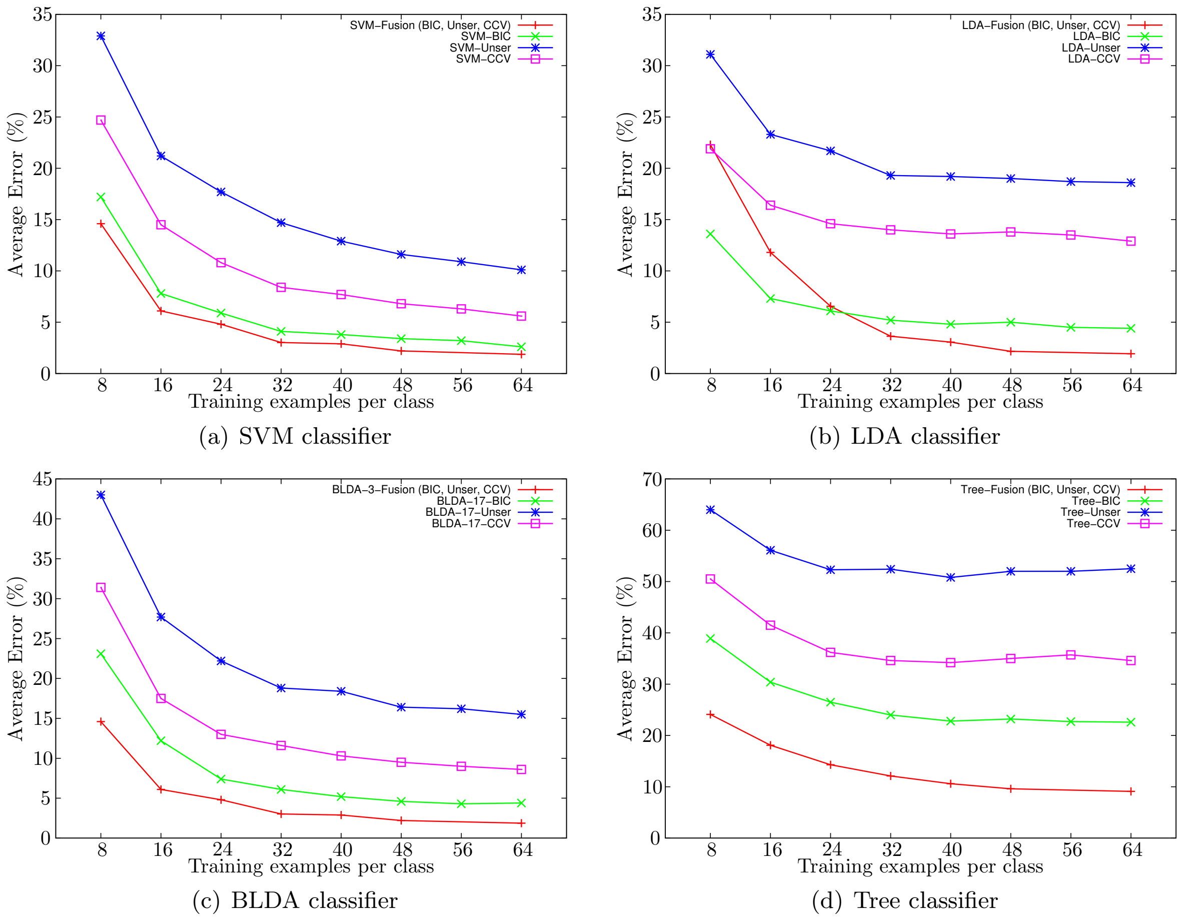
Average error per class

One important aspect when dealing with classification is the average error, or its counterpart, average expected accuracy per class. This information, shows which class of our data needs more attention when solving the confusions. Figure 5.15 shows us the average expected error for each one of our 15 classes. Clearly, we see that *Fuji apple* is one class that needs particular attention. It yields the highest error when compared to the other classes. Another class the has an interesting error behavior is *Onions*. After the error decreases when using up to 40 training examples it becomes higher as the number of training examples increases.

This experiments shows that for some classes, it might be worth performing the training with less examples. Indeed, this is possible when using our fusion approach, since we analyze each pair of classes with a separate classifier.

Average time

The average time for the image descriptors feature extraction and classification using any of the classifiers under our feature and classifier fusion approach still is less than 1 second. However, the more examples in the training set the more time consuming are the combinations in the training stage. For instance, to train a multi-class classifier using our approach with SVM classifier, 48



(a) SVM classifier

(b) LDA classifier

(c) BLDA classifier

(d) Tree classifier

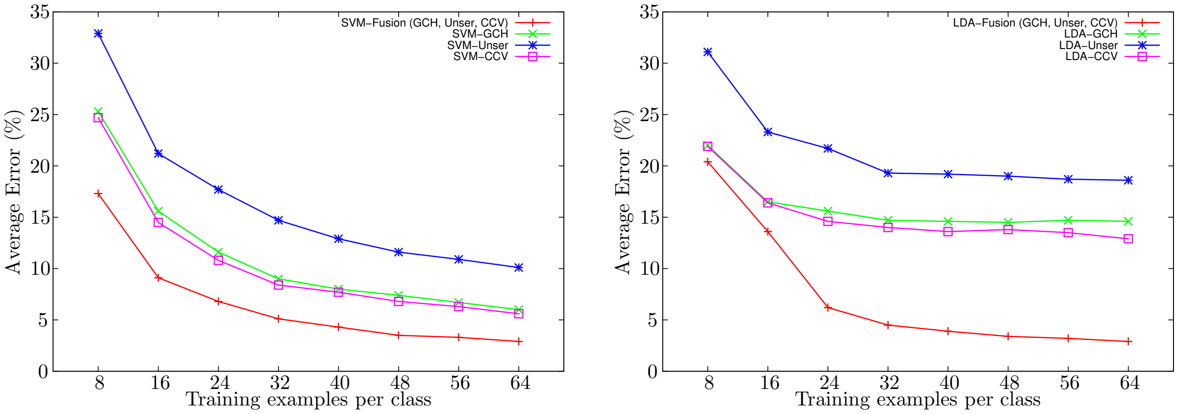
Figure 5.13: Average error results for fusion of BIC, CCV, and Unser features.

training examples per class, and the combination of the features BIC, CCV, and Unser, it is necessary about one hour in one 2.1GHz machine with 2GB of RAM.

Finally, the use of complex appearance descriptors such as appearance part-based ones can impact the training set without yielding significant improvements on the overall effectiveness of the classification system.

5.8 Conclusions and remarks

To solve a complex problem such us the one in this paper using just one feature descriptor is a difficult task. We discussed that although normal feature fusion is quite effective for some problems, it can yield unexpected classification results when not properly normalized and prepared. Besides it has the drawback of increasing the dimensionality which might require more training



(a) SVM classifier

(b) LDA classifier

Figure 5.14: Average error results for fusion of GCH, CCV, and Unser features.

data.

Therefore, we proposed to cope with the multi-class problem as a set of binary problems in such a way we can fuse diverse features and classifier approaches specialized to parts of the problem. We presented a unified solution that can combine many features and classifiers that requires less training and performs better if compared with a naïve method, where all features are simply concatenated and fed independently to each classification algorithm. In addition, the proposed solution is amenable to continuous learning, both when refining a learned model and also when adding new classes to be discriminated.

A second contribution of our work is the presentation of a complete and well-documented fruit/vegetables image data set suitable for content-based image retrieval, object recognition, and image categorization tasks. We hope this data set will endure beyond this paper as a common comparison set for researchers working in this space.

Table 5.2 counterpoints the main aspects of our proposed approach with respect to the naïve binding of features in a big feature vector. We point out advantages and drawbacks of each approach and leave the decision of using one or another to the designer of a classification system. The first and foremost aspect of both approaches is that with a certain minimum number of training examples, when all data can be properly normalized, and if we do not face the dimensionality problem, their results are comparable.

Another observation is that for both approaches, it seems to be not advisable to combine weak features with high classification errors and features with low classification errors. In this case, most likely the system will not take advantage of such combination.

Whether or not more complex approaches such as appearance-based descriptors provide good results for the classification is still an open problem. It would be unfair to conclude they do not help in the classification given that, their success is highly based on their patches representation. Nevertheless, it is fact that such approaches are computational demanding and

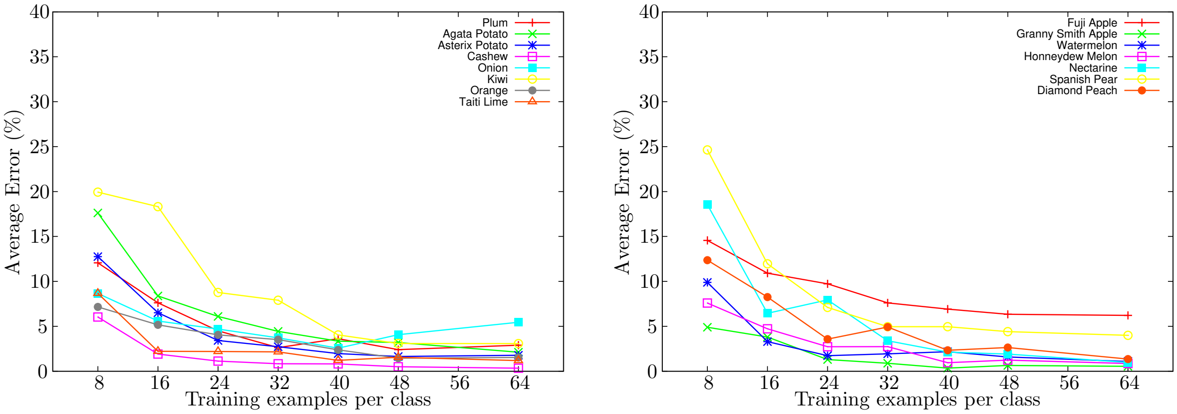


Figure 5.15: Average error per class using fusion of the features BIC, CCV, and Unser, for an SVM classifier.

perhaps not advisable in some scenarios.

Further work include the improvement of the fruits/vegetables representative patches, and the analysis of other appearance and texture image descriptors. Furthermore, we are interested in the incorporation of spatial constraints among the local descriptors. In addition, we want to create the conditions for a semi-supervised approach that leads us to continuous learning, taking advantage of misclassified examples. In a semi-supervised scenario, the initial training stage can be simplified requiring only a few examples for each analyzed class.

Acknowledgments

We are grateful to the people at our local fruits and vegetables distribution center for their patience and help. In special, we thank Francisco Domingos of Agro Comercial Bella Vista Ltda. Finally, we thank FAPESP (Grants 05/58103-3 and 07/52015-0) and CNPq (Grants 309254/2007-8 and 551007/2007-9) for their financial support.

Feature binding fusion	**Feature and classifier fusion using multi-class from binary**
Short definition. It performs feature fusion by assembling together (binding) the diverse feature vectors resulting in a big feature vector.	**Short definition**. It performs feature and classifier fusion by considering the multi-class problem as a set of binary problems and using specialized binary classifiers tuned for specific parts (divide-and-conquer).
Dimensionality problem. It might face the dimensionality problem when binding many features and creating a big feature vector.	**Dimensionality problem**. Less prone to the dimensionality problem in the feature fusion because it uses each feature independently.
Selection of best features. Given that all the features are addressed as just one feature, it is hard to point out which ones are more effective in the fusion.	**Selection of best features**. It is straightforward to point out which features are effective in the fusion.
Training. New features and classes usually require a complete new training. In some cases, due to the dimensionality problems, adding new features might require more training data.	**Training**. New features and classes would require partial training. It only needs to train new binary classifiers for the new features and for the new classes compared to the previous ones. Usually, more features do not require more training data.
Data normalization. It requires proper normalization of the data. Once new features are added, a new normalization and retraining must be performed.	**Data normalization**. It does not require special normalization given that each feature is independently analyzed.
Testing. If the normalization is possible and properly handled, the classification is easy.	**Testing**. The classification requires more storage for the classifier parameters. Furthermore, the classification itself is more time consuming.
Combining classifiers. The classifier is tied to the features. It is not straightforward to use different classifiers for different features.	**Combining classifiers**. Different classifiers and features can be combined with no restrictions.

Table 5.2: Comparison between feature binding fusion and feature/classifier fusion using multi-class from binary.

Multi-classe a Partir de Classificadores Binários

Muitos problemas reais de reconhecimento e de classificação frequentemente necessitam mapear várias entradas em uma dentre centenas ou milhares de possíveis categorias.

Muitos pesquisadores têm proposto técnicas efetivas para classificação de duas classes nos últimos anos. No entanto, alguns classificadores poderosos tais como SVMs são difíceis de estender para o cenário multi-classe. Em tais casos, a abordagem mais comum é a de reduzir a complexidade do problema multi-classe para pequenos e mais simples problemas binários (dividir para conquistar).

Ao utilizar classificadores binários com algum critério final de combinação (redução de complexidade), muitas abordagens descritas na literatura partem do princípio de que os classificadores binários utilizados na classificação são independentes e aplicam um sistema de votação como política final de combinação. Entretanto, a hipótese da independência não é a melhor escolha em todos os casos.

Nesse sentido, nos deparamos com um problema interessante: como tornar a utilização de poderosos classificadores binários no contexto multi-classe mais eficiente e eficaz?

No Capítulo 6, introduzimos uma técnica para combinação de classificadores binários (chamados classificadores base) para a resolução de problemas no contexto geral de multi-classificação. Nós denominamos a técnica de *Affine-Bayes.* Finalmente, mostramos que nossa solução torna possível resolver problemas complexos tais como de 100 ou 1000 classes a partir da combinação de poucos, mas poderosos, classificadores binários.

O Capítulo 6 é uma compilação de nosso trabalho submetido à *IEEE Transactions on Pattern Analysis and Machine Intelligence* (TPAMI) e do artigo [150] no *Intl. Conference on Computer Vision Theory and Applications* (VISAPP).

Chapter 6

From Binary to Multi-class: A Bayesian Evolution

Abstract

Recently, there has been a lot of success in the development of effective binary classifiers. Although many statistical classification techniques do have natural multi-class extensions, some, such as the SVMs, do not. Therefore, it is important to know how to map the multi-class problem into a set of simpler binary classification problems. In this paper, we introduce a new Bayesian multi-class from binary reduction method: the *Affine-Bayes Multi-class*. First, we introduce the concept of affine relations among the binary classifiers (dichotomies), and then we present a principled way to find groups of highly correlated base learners. With that in hand, we can learn the proper joint probabilities that allow us to predict the class. Finally, we present two additional strategies: one to reduce the number of required dichotomies in the multi-class classification and the other to find new dichotomies in order to replace the less discriminative ones. We can use these two new procedures iteratively to complement the base *Affine-Bayes Multi-class* and boost the overall multi-class classification performance. We validate and compare our approach to the literature in several open datasets that range from small (10 to 26 classes) to large multi-class problems (1,000 classes) always using simple reproducible descriptors.

6.1 Introduction

Supervised learning is a Machine Learning strategy to create a prediction function from training data. The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of domain-related training examples [16]. Many supervised learning techniques are conceived for binary classification [127]. However, a lot of real-world recognition and classification problems often require that we map inputs to one out of hundreds or thousands of possible categories.

Several researchers have proposed effective approaches for binary classification in the last years. Successful examples of such approaches are margin and linear classifiers, decision trees, and ensembles. We can easily extend some of those techniques to multi-class problems (e.g., decision trees). However, we can not easily extend to multi-class some others powerful and popular classifiers such as SVMs [31]. In such situations, the usual approach is to reduce the multi-class problem complexity into multiple simpler binary classification problems. Binary classifiers are more robust to the curse of dimensionality than multi-class approaches. Hence, it is worth dealing with a larger number of binary problems.

A *class binarization* is a mapping of a multi-class problem onto several two-class problems (divide-and-conquer) and the subsequent combination of their outcomes to derive the multi-class prediction [130]. We refer to the binary classifiers as *base learners* or *dichotomies*.

There are many possible approaches to reduce multi-class to binary classification problems. We can classify such approaches into three broad groups [145]: (1) *One-vs-All* (OVA), (2) *One-vs-One* (OVO), and (3) *Error Correcting Output Codes* (ECOC). Also, the multi-class decomposition into binary problems usually contains three main parts: (1) the ECOC matrix creation; (2) the choice of the base learner; and (3) the decoding strategy.

Our focus here is on the creation of the ECOC matrix and on the decoding strategy. Although we shall explain latter, for the creation of the ECOC matrix, it is important to choose a feasible number of dichotomies to use. In general, the more base learners we use, the more complex is the overall procedure. For the decoding strategy, it is essential to choose a deterministic strategy robust to ties and errors in the dichotomies' prediction.

In this paper, we introduce a brand new way to combine binary classifiers to perform large multi-class classification. We present a new Bayesian treatment for the decoding strategy, the *Affine-Bayes Multi-class*.

We propose a decoding approach based on the conditional probabilities of groups of high-correlated binary classifiers. For that, we introduce the concept of affine relations among binary classifiers and present a principled way to find groups of high correlated dichotomies.

Furthermore, we introduce two additional strategies: one to reduce the number of required dichotomies in the multi-class classification and the other to find new dichotomies in order to replace the less discriminative ones. These two new procedures are optional and can iteratively complement the base *Affine-Bayes Multi-class* to boost the overall multi-class classification performance.

Contemporary Vision and Pattern Recognition problems such as face recognition, fingerprinting identification, image categorization, and DNA sequencing often have an arbitrarily large number of classes to cope with. Finding the right descriptor is just the first step to solve a problem. Here, we show how to use a small number of simple, fast, and weak or strong base learners to achieve good results, no matter the choice of the descriptor. This is a relevant issue for large-scale classification problems.

We validate our approach using data sets from the UCI repository, NIST digits, Corel Photo

Gallery, and the Amsterdam Library of Objects. We show that our approach provides better results than all comparable approaches in the literature. Furthermore, we also compare our approach to Passerini et al. [127], who proposed a Bayesian treatment for decoding assuming independence among all binary classifiers.

We organize this paper as follows. In Section 6.2, we outline the state-of-the-art in multi-class classification. In Section 6.3, we present our new Bayesian treatment for the decoding strategy: the *Affine-Bayes Multi-class* as well as two new strategies to boost performance: the *Shrinking* and *Augmenting* stages. Section 6.4 presents our experiments and results. In Section 6.5, we draw conclusions and point out some future directions. Finally, in the Appendix, we provide a table of symbols.

6.2 State-of-the-Art

Most of the existing literature addresses one or more of the three main parts of a multi-class decomposition problem: (1) the ECOC matrix creation; (2) the dichotomies choice; and (3) the decoding.

In the following, let $\mathcal{T}$ be the team (set) of used dichotomies $\mathcal{D}$ in a multi-class problem, $N_{\mathcal{T}}$ be the size of $\mathcal{T}$, and N_c be the number of classes[1].

There are three broad groups for reducing multi-class to binary: *One-vs-All*, *One-vs-One*, and *Error Correcting Output Codes* based methods [130].

1. **One-vs-All (OVA)**. Here, we use $N_{\mathcal{T}} = N_c = O(N_c)$ binary classifiers (dichotomies) [5, 28]. We train the i^{th} classifier using all patterns of class i as positive $(+1)$ examples and the remaining class patterns as negative (-1) examples. We classify an input example x to the class with the highest response.

2. **One-vs-One (OVO)**. Here, we use $N_{\mathcal{T}} = \binom{N_c}{2} = O(N_c^2)$ binary classifiers. We train the ij^{th} dichotomy using all patterns of class i as positive and all patterns of class j as negative examples. In this framework, there are many approaches to combine the obtained outcomes such as *voting*, and *decision directed acyclic graphs* (DDAGs) [136].

3. **Error Correcting Output Codes (ECOC)**. Proposed by Dietterich and Bakiri [38], in this approach, we use a coding matrix $M \in \{-1, 1\}^{N_c \times N_{\mathcal{T}}}$ to point out which classes to train as positive and negative examples. Allwein et al. [4] have extended such approach and proposed to use a coding matrix $M \in \{-1, 0, 1\}^{N_c \times N_{\mathcal{T}}}$. In this model, the j^{th} column of the matrix induces a partition of the classes into two meta-classes. An instance x belonging to a class i is a positive instance for the j^{th} dichotomy if and only if $M_{ij} = +1$. If $M_{ij} = 0$, then it indicates that the i^{th} class is not part of the training of the j^{th} dichotomy. In this framework, there are many approaches to combine the obtained outcomes such as *voting*, *Hamming* and *Euclidean distances*, and *loss-based functions* [192].

[1] In the Appendix, we provide a table of symbols.

> When the dichotomies are margin-based learners, Allwein et al. [4] have showed the advantage and the theoretical bounds of using a loss-based function of the margin. Klautau et al. [82] have extended such bounds to other functions.

It is worth noting that the asymptotic complexity $O(N_c)$ refers to the number of required dichotomies to perform a classification. It is not the only measure used to calculate the time required for training and testing.

For training, we need to consider the number of training examples of each dichotomy, the number of dichotomies used, and the complexity of each binary classifier with respect to the number of minimum required operations to perform a classification. For testing, we need to consider the number of dichotomies used and the complexity of each binary classifier.

In the case of OVO, the time complexity refers to the $O(N_c^2)$ dichotomies, each one requiring positive and negative examples of the two classes being trained each time. For the testing, OVO requires $O(N_c^2)$ binary classifications for each tested instance.

In the case of OVA, the time complexity refers to the $O(N_c)$ dichotomies, each one requiring positive examples of the class of interest and the negative examples of all the remaining classes. For the testing, OVA requires $O(N_c)$ binary classifications for each tested instance.

Pedrajas et al. [130] have proposed to combine the strategies of OVO and OVA. Although the combination improves the overall multi-class effectiveness, the proposed approach uses $N_{\mathcal{T}} = \binom{N_c}{2} + N_c = O(N_c^2)$ dichotomies in the training stage. Moreira and Mayoraz [112] also developed a combination of different classifiers. They have considered the output of each dichotomy as a probability of the pattern of belonging to a given class. This method requires $\frac{N_c(N_c+1)}{2} = O(N_c^2)$ base learners. Athisos et al. [7] have proposed class embeddings to choose the best dichotomies from a set of trained base learners.

Pujol et al. [145] have presented a heuristic method for learning ECOC matrices based on a hierarchical partition of the class space that maximizes a discriminative criterion. The proposed technique finds the potentially best $N_c - 1 = O(N_c)$ dichotomies to the classification. Crammer and Singer [33] have proven that the problem of finding optimal discrete codes is NP-complete. Hence, Pujol et al. have used a heuristic solution for finding the best candidate dichotomies. Even such solution is computationally expensive, and the authors only report results for $N_c \leq 28$.

Takenouchi and Ishii [173] have used the information transmission theory to combine ECOC dichotomies. The authors use the full coding matrix M for the dichotomies, i.e., $N_{\mathcal{T}} = \frac{3^{N_c} - 2^{N_c+1} + 1}{2} = O(3^{N_c})$ dichotomies. The authors only report results for $N_c \leq 7$ classes.

Young et al. [195] have used dynamic programming to design an one-class-at-a-time removal sequence planning method for multi-class decomposition. Although their approach only requires $N_{\mathcal{T}} = N_c - 1$ dichotomies in the testing phase, the removal policy in the training phase is expensive. The removal sequence for a problem with N_c classes is formulated as a multi-stage decision-making problem and requires $N_c - 2$ classification stages. In the first stage, the method uses N_c dichotomies. In each one of the $N_c - 3$ remaining stages, the method uses $\frac{N_c(N_c-1)}{2}$

dichotomies. Therefore, the total number of required base learners are $\frac{N_c^3-4N_c^2+5N_c}{2} = O(N_c^3)$.

Passerini et al. [127] have introduced a decoding function that combines the margins through an estimation of their class conditional probabilities. The authors have assumed that all base learners are independent and solved the problem using a Naïve Bayes approach. Their solution works regardless of the number of selected dichotomies and can be associated with each one of the previous approaches.

6.3 Affine-Bayes Multi-class

In this section, we present our new Bayesian treatment for the decoding strategy: the *Affine-Bayes Multi-class*. We propose a decoding approach based on the conditional probabilities of groups of affine binary classifiers. For that, we introduce the concept of affine relations among binary classifiers, and present a principled way to find groups of high correlated dichotomies. Finally, we present two additional strategies: one to reduce the number of required dichotomies in the multi-class classification and the other to find new dichotomies in order to replace the less discriminative ones. We can use these two new procedures iteratively to complement the base *Affine-Bayes Multi-class* to boost the overall multi-class classification performance.

To classify an input, we use a team of trained base learners $\mathcal{T}$. We call $\mathcal{O}_{\mathcal{T}}$ a realization of $\mathcal{T}$. Each element of $\mathcal{T}$ is a binary classifier (dichotomy) and produces an output $\in \{-1, +1\}$. Given an input element x to classify, a realization $\mathcal{O}_{\mathcal{T}}$ contains the information to determine the class of x. In other words, $P(y = c_i|x) = P(y = c_i|\mathcal{O}_{\mathcal{T}})$.

However, we do not have the probability $P(y \;=\; c_i \mid \mathcal{O}_{\mathcal{T}})$. From Bayes theorem,

$$\begin{aligned} P(y &= c_i|\mathcal{O}_{\mathcal{T}}) = \frac{P(\mathcal{O}_{\mathcal{T}}|y = c_i)P(y = c_i)}{P(\mathcal{O}_{\mathcal{T}})} \\ &\propto P(\mathcal{O}_{\mathcal{T}}|y = c_i)P(y = c_i) \end{aligned} \tag{6.1}$$

$P(\mathcal{O}_{\mathcal{T}})$ is just a normalizing factor and it is suppressed.

Previous approaches have solved the above model by considering the independence of the dichotomies in the team $\mathcal{T}$ [127]. If we consider independence among all dichotomies, the model in Equation 6.1 becomes

$$P(y = c_i|\mathcal{O}_{\mathcal{T}}) \propto \prod_{t \,\in\, \mathcal{T}} P(\mathcal{O}_{\mathcal{T}}^t|y = c_i)P(y = c_i), \tag{6.2}$$

and the class of the input x is given by

$$cl(x) \;=\; \arg\max_i \prod_{t \,\in\, \mathcal{T}} P(\mathcal{O}_{\mathcal{T}}^t|y = c_i)P(y = c_i). \tag{6.3}$$

Although the independence assumption simplifies the model, it comes with limitations and it is not the best choice in all cases [114]. In general, it is quite difficult to handle independence without using smoothing functions to deal with numerical instabilities when the number of terms in the series is too large. In such cases, it is necessary to find a suitable density distribution to describe the data, making the solution more complex.

We relax the assumption of independence among all binary classifiers. When two of these dichotomies have a lot in common, it would be unwise to threat their results as independent random variables (RVs). In our approach, we find groups of affine classifiers (high correlated dichotomies) and represent their outcomes as dependent RVs, using a single *conditional probability table* (CPT) as an underlying distribution model. Each group then has its own CPT, and we combine the groups as if they are independent from each other — to avoid a dimensionality explosion.

Our technique is a Bayesian-Network-inspired approach for RV estimation. We decide the RV that represents the class based on the RVs that represent the outcomes of the dichotomies.

We model the multi-class classification problem conditioned to groups of affine dichotomies $\mathcal{G}_{\mathcal{D}}$. The model in Equation 6.1 becomes

$$P(y = c_i | \mathcal{O}_{\mathcal{T}}, \mathcal{G}_{\mathcal{D}}) \propto P(\mathcal{O}_{\mathcal{T}}, \mathcal{G}_{\mathcal{D}} | y = c_i) P(y = c_i). \tag{6.4}$$

We assume independence only among the groups of affine dichotomies $g_i \in \mathcal{G}_{\mathcal{D}}$. Therefore, the class of an input x is given by

$$cl(x) = \arg\max_j \prod_{g_i \, \in \, \mathcal{G}_{\mathcal{D}}} P(\mathcal{O}_{\mathcal{T}}^{g_i}, g_i | y = c_j) P(y = c_j). \tag{6.5}$$

To find the groups of affine classifiers $\mathcal{G}_{\mathcal{D}}$, we define an affinity matrix $\mathcal{A}$ among the classifiers. The affinity matrix measures how affine are two dichotomies when classifying a set of training examples X. In Section 6.3.1, we show how to create the affinity matrix $\mathcal{A}$. After calculating the affinity matrix $\mathcal{A}$, we use a clustering algorithm to find the groups of correlated binary classifiers in $\mathcal{A}$. In Section 6.3.2, we show how to find the groups of affine dichotomies from an affinity matrix $\mathcal{A}$.

The groups of affine classifiers can contain classifiers that do not contribute significantly to the overall classification. Therefore, we can deploy a procedure to identify the less important dichotomies within an affine group and eliminate them.

With this shrinking stage, we can have two different objectives. On one hand, we might want just to reduce the number of required dichotomies to perform the multi-class classification and hence speed-up the overall process and make robust CPTs estimations. On the other hand, we might want to eliminate less discriminative classifiers in order to replace them with more powerful ones.

In Section 6.3.3, we show a consistent approach to eliminate the less important dichotomies within an affine group. In addition, in Section 6.3.4, we introduce a simple idea to find dichotomies to replace the ones tagged as less discriminative in the *Shrinking* stage.

We can apply the *Shrinking* and *Augmenting* procedures iteratively until a convergence criterion is satisfied. These two procedures are very fast since most of the information they need is already calculated during the earlier training.

In Algorithm 7, we present the main steps of our model for multi-class classification. In line 1, we divide the training data into five parts and use four parts to train the dichotomies and one part to validate the trained dichotomies and to construct the conditional probability tables[2]. In lines 3–6, we train and validate each dichotomy using a selected method. The method can be any binary classifier such as LDA, or SVM.

Each dichotomy produces an output $\in \{-1, +1\}$ for each input x. In line 8, $\mathcal{O}$ contains all realizations of the available dichotomies for the input data X'. In lines 10 and 11, we find groups of affine dichotomies using the realization $\mathcal{O}_i$.

Using the information of groups of affine dichotomies, in line 12, we create a CPT for each affine group. These CPTs provide the joint probabilities of a realization $\mathcal{O}_{\mathcal{T}}$ and the affine groups $g_i \subset \mathcal{G}_{\mathcal{D}}$ when testing an unseen input data x.

In line 13, our approach verifies if the user wants to find the less discriminative dichotomies and replace them with better ones iteratively. If so, in line 14, we perform the shrinking stage in order to tag the best dichotomies in the multi-class process.

In line 15, we use the error calculated in the training to find new dichotomies. In line 16, we train the new dichotomies found in line 15. Note that in each iteration there are only a few of them.

Afterwards, in line 17, we update the affinity matrix $\mathcal{A}$ to consider only the best dichotomies tagged in the shrinking stage and, naturally, the new ones produced by the augmenting procedure. We also update the realizations $\mathcal{O}^i$ and the conditional probabilities accordingly. In line 18, we perform clustering on the affinity matrix and find the updated groups of representative dichotomies $\mathcal{G}_{\mathcal{D}}'$.

In line 19, we repeat lines 14–18 until a convergence criterion is satisfied. In our case, if the training error produced by the updated team of classifiers is bigger than the error of the previous step, we stop the iteration and discard the proposed augmented dichotomies. This is a simple criterion and, as we show in the experiments, it yields good results.

If the user does not want to iteratively find the dichotomies, she might be interested in the reduced set of dichotomies. Therefore, line 20 finds the best dichotomies within the affine groups. This information can be used in the testing phase, for instance, simply to reduce the number of used dichotomies.

6.3.1 Affinity matrix $\mathcal{A}$

Given a training data set X, we introduce a metric to find the affinity between two dichotomies realizations $\mathcal{D}_i$, $\mathcal{D}_j$ whose outputs $\in \{-1, +1\}$

$$\mathcal{A}_{i,j} = \frac{1}{N} \left| \sum_{\forall\ x\ \in\ X} \mathcal{D}_i(x)\mathcal{D}_j(x) \right|, \forall\ \mathcal{D}_i \text{ and } \ \mathcal{D}_j \in \mathcal{T}. \tag{6.6}$$

[2] The cross-validation is not a required step to our approach. We perform cross-validation in order to provide fair results across the data sets.

Algorithm 7 Affine-Bayes Multi-class.

```
Require: Training data set $X$, Testing data $X^t$, a team of binary classifiers $\mathcal{T}$, toggle parameter
    $augment \in \{true, false\}$
 1: Split $X$ into $k$ parts, $X_i$ such that $i = 1 \dots k$;
 2: for each $X_i$ do                                              ▷ Inner $k$-fold cross-validation.
 3:     $X' \leftarrow X \setminus X_i$;
 4:     for each dichotomy $d \in \mathcal{T}$ do
 5:         $D_{train} \leftarrow \textsc{Train}(X', d, method)$;
 6:         $\mathcal{O}_d^i \leftarrow \textsc{Test}(X_i, d, method, D_{Train})$;
 7:     end for
 8:     $\mathcal{O}^i \leftarrow \bigcup(\mathcal{O}_d^i)$;
 9: end for
10: Create the affinity matrix $\mathcal{A}$ for $\bigcup \mathcal{O}^i$;
11: Perform clustering on $\mathcal{A}$ to find the affine groups of dichotomies $\mathcal{G}_{\mathcal{D}}$;
12: Create a CPT for each group $g \subset \mathcal{G}_{\mathcal{D}}$ of affine dichotomies using $\mathcal{O}$;
13: if $augment = true$ then                                       ▷ Shr/Aug desired
14:     Perform shrinking. $\mathcal{GS}_D \leftarrow \textsc{Shrink}(\mathcal{G}_{\mathcal{D}})$;
15:     Perform augmenting. $\mathcal{TA}_D \leftarrow \textsc{Augment}(D_{train})$;
16:     Perform the training in lines 2–7 only for the new dichotomies $\mathcal{TA}_D$ and store the result in
            $D_{train}^{aug}$.
17:     Update $\mathcal{O}^i, \mathcal{A}$ and the CPT to consider the representative elements tagged in the shrinking and
            the new ones produced in the augmenting.
18:     $\mathcal{G}'_{\mathcal{D}} \leftarrow$ clustering of the updated matrix $\mathcal{A}$
19:     Repeat lines 14–18 while the convergence is not satisfied.
20: else Perform shrinking. $\mathcal{GS}_D \leftarrow \textsc{Shrink}(\mathcal{G}_{\mathcal{D}})$;
21: end if
22: for each $x \in X^t$ do
23:     Perform the classification of $x$ from the model on Equation 6.5 either using the set of affine
            dichotomies $\mathcal{G}_{\mathcal{D}}$, the shrinked $\mathcal{GS}_{\mathcal{D}}$, or the optimized $\mathcal{G}'_D$.
24: end for
```

According to the affinity model, if two dichotomies have the same output for all elements in X, their affinity is 1. For instance, this is the case when $\mathcal{D}_i = \mathcal{D}_j$. If $\mathcal{D}_i \neq \mathcal{D}_j$ in all cases, their affinity is also 1. On the other hand, if two dichotomies have half outputs different and half equal, their affinity is 0. Using this model, we can group binary classifiers that produce similar outputs and, further, eliminate those which do not contribute significantly to the overall classification procedure.

6.3.2 Clustering

Given an affinity matrix $\mathcal{A}$ representing the relationships among all dichotomies in a team $\mathcal{T}$, we want to find groups of classifiers that have similar affinities. We strive for finding groups of dependent classifiers while the groups are independent from one another. A good clustering approach is important to provide balanced groups of dichotomies. Balancing is interesting because it leads to simpler conditional probability tables.

As noted by Ben-Hur et al. [14], we often regard clusters as continuous concentration of data points. Two points belong to a cluster if they are close to each other, or if they are well connected by paths of short "hops" over other points. The more we have such paths, the higher are the chances the points belong to a cluster [47]. In this sense, Fisher and Poland [47], have introduced a spectral clustering approach which we use in this paper. Instead of considering two points similar if they are connected by a high-weight edge, the authors propose assign them a high correlation if the overall graph conductivity between them is high. These considerations exhibit an analogy to electrical networks: the conductivity between two nodes depends not only on the conductivity of the direct path between them, but also on other indirect paths.

To find the conductivity for any two points $\boldsymbol{x_p}$ and $\boldsymbol{x_q}$, we first solve the system of linear equations:

$$\boldsymbol{G\varphi = \eta}, \tag{6.7}$$

where $\boldsymbol{G}$ is a matrix constructed from the original affinity matrix $\boldsymbol{\mathcal{A}}$:

$$\boldsymbol{G(p,q)} = \begin{cases} \text{for p = 1:} & \begin{cases} \boldsymbol{1} & \text{for } \boldsymbol{q = 1} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \\ \text{otherwise:} & \begin{cases} \sum_{\boldsymbol{k \neq p}} \boldsymbol{\mathcal{A}(p,q)} & \text{for } \boldsymbol{p = q} \\ \boldsymbol{-\mathcal{A}(p,q)} & \text{otherwise} \end{cases} \end{cases} \tag{6.8}$$

and $\boldsymbol{\eta}$ is the vector representing points for which the conductivity is computed:

$$\boldsymbol{\eta(k)} = \begin{cases} \boldsymbol{-1} & \text{for } \boldsymbol{k = p} \text{ and } \boldsymbol{p > 1} \\ \boldsymbol{1} & \text{for } \boldsymbol{k = q} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{6.9}$$

Then the conductivity between $\boldsymbol{x_p}$ and $\boldsymbol{x_q}$, $\boldsymbol{p < q}$, due to the way $\boldsymbol{\eta}$ is constructed, is given by

$$\boldsymbol{C(p,q) \;=\; \left(G^{-1}(p,p) + G^{-1}(q,q) - G^{-1}(p,q) - G^{-1}(q,p)\right)^{-1}}. \tag{6.10}$$

Due to symmetry, $\boldsymbol{C(p,q) = C(q,p)}$ and it is necessary to compute $\boldsymbol{G^{-1}}$ only once. The conductivity matrix $\boldsymbol{C}$ can be computed in $\boldsymbol{O(N^2)}$ time. After building the conductivity matrix from $\boldsymbol{\mathcal{A}}$, we can perform the clustering using any simple cluster method such as klines as proposed in [47]. Although Fisher and Poland's algorithm [47] has provided very good results for our problem, we also have observed similar results using a simpler, yet effective, approach with a greedy algorithm for finding the dependent groups of dichotomies from the affinity matrix.

In the greedy clustering approach, first we find the dichotomy with the highest affinity sum with respect to all its neighbors (row with highest sum in $\boldsymbol{\mathcal{A}}$). After that, we select the neighbors with affinity greater or equal than a threshold $\boldsymbol{t}$. Next, we check if each dichotomy in the group is affine to the others and select those satisfying this requirement. This procedure results the first affine group. Afterwards, we remove the selected dichotomies from the main team $\boldsymbol{\mathcal{T}}$ and repeat the process until we analyze all available dichotomies. Throughout experiments, we have found that $\boldsymbol{t = 0.6}$ is a good threshold.

6.3.3 Shrinking

Sometimes, when modeling a problem using conditional probabilities, we have to deal with large conditional probability tables which can lead to over-fitting. One approach to cope with this problem is to suppose independence among all dichotomies which results in the smallest possible CPT. However, as we show in this paper, this approach limits the representative power of the Bayes approach. In the following, we show an alternative approach.

In the shrinking stage, we want to find the dichotomies within a group that are more relevant for the overall multi-class classification. For that, we find the accumulative entropy of each classifier within a group from the examples in the training data $\boldsymbol{X}$. The higher the accumulative entropy, the more representative is a specific dichotomy. Let $\boldsymbol{h_{ij}}$ be the accumulative entropy for the classifier $\boldsymbol{j}$ within a group of affine dichotomies $\boldsymbol{i}$. We define $\boldsymbol{h_{ij}}$ as

$$\boldsymbol{h_{ij} = \sum_{c \in C_L} \sum_{x \in X} \left(p_c^x \log_2(p_c^x) + (1 - p_c^x) \log_2 (1 - p_c^x) \right)} \tag{6.11}$$

where $\boldsymbol{p_c^x = P(y = c \mid x, g_i^j, \mathcal{O}_x^{g_i^j})}$, $\boldsymbol{g_i^j}$ is the $\boldsymbol{j^{th}}$ dichotomy within the affine group $\boldsymbol{g_i}$, $\boldsymbol{\mathcal{O}_x^{g_i^j}}$ is its realization for the input $\boldsymbol{x}$, and $\boldsymbol{c \in C_L}$ the available class labels.

We choose the classifiers with the highest cumulative entropy to select the best classifiers within an affine group. We have found in the experiments, that selecting $\boldsymbol{60\%}$ of the classifiers is a good tradeoff between multi-class overall effectiveness and efficiency. One could use another cutting criterion, such as the maximum CPT size. It is worth noting that this procedure is very fast once we already have the required probabilities stored in the previously computed CPTs. Hence, we do not need to scan the data to perform it.

During the training phase, our approach finds the affine groups of binary classifiers and tags the most relevant dichotomies within each group. On one hand, this information can be used in association with the *Augmenting* step (c.f., Sec. 6.3.4) that finds new substitutes for the non-selected dichotomies. Both procedures, *Shrinking* and *Augmenting* can be performed until a convergence criterion is satisfied. On the other hand, if no optimization for finding replacements for the dichotomies is intended, the selected group of dichotomies can be used in the testing phase simply to reduce the number of required classifiers in the multi-class task.

In summary, with our solution, we measure the affinity on the training data to learn the binary classifiers relationship and decision surface. It is a simple and fast way to estimate the distribution. Sometimes, a dichotomy may be in the *team* because it is critical for discriminating between two particular classes. If so, it is unlikely it will share a group of high-correlated classifiers because it would require this dichotomy to be high-correlated with all dichotomies in such group. We have performed some experiments to test that and, in all tested cases, such dichotomies specific for rare classes are kept in the final pool of dichotomies.

6.3.4 Augmenting

As we showed in Section 6.3.3, we are able to find and tag the dichotomies that are more relevant for the overall multi-class performance with a simple approach. This leads us to the natural consequence of finding new dichotomies to replace the ones that are less discriminative. For this intent, in the augmenting stage, we want to find new dichotomies (rather than simple random choice) to replace the less representative ones eliminated in the shrinking stage in order to improve the overall multi-class performance.

To perform the augmenting, we use the confusion matrix generated in the training stage. From such confusion matrix, we are able to point out the classes that are in more confusion with each other. We use such information to build a representation of the confusions through hierarchical clustering in order to determine the ones that need urgent dichotomies to solve them. Afterwards, we sort the clusters according to the sum of the number of confusions normalized by by the number of edges connecting the elements in that cluster excluding the self-references. For each cluster, in order, we apply normalized cuts [165] to find the cut that maximizes the separability of the cluster and, therefore, its confusion.

We summarize these procedures in Algorithm 8.

Algorithm 8 Augmenting procedure for Affine-Bayes.

```
Require: A confusion matrix $\boldsymbol{\mathcal{C}}$ already calculated in the training stage, and the number of dichotomies
    $\boldsymbol{n}$ to generate
Set the diagonal elements of $\boldsymbol{\mathcal{C}}$ to zero
$\boldsymbol{\mathcal{C}_{ij}} \longleftarrow \boldsymbol{1 - \mathcal{C}_{ij} / \sum_{i,j} \mathcal{C}_{ij}} \;\; \forall \boldsymbol{\mathcal{C}_{ij}} \in \boldsymbol{\mathcal{C}}$        ▷ Normalizing $\boldsymbol{\mathcal{C}}$
$\boldsymbol{H}$ ⟵ hierarchical clustering of $\boldsymbol{\mathcal{C}}$ using, for instance, the simple Agnes [78] algorithm.
Sort $\boldsymbol{H}$ according to the sum of the confusions of each group $\boldsymbol{h_i} \subseteq \boldsymbol{H}$ normalized by the number of
        edges connecting the elements in the group $\boldsymbol{h_i}$ excluding the self-references.
for each group $\boldsymbol{h_i} \subseteq \boldsymbol{H}$ do
    $\boldsymbol{d}$ ⟵ normalized cuts of $\boldsymbol{h_i}$
    $\boldsymbol{\mathcal{T}_A} \longleftarrow \boldsymbol{\mathcal{T}_A} \cup \boldsymbol{d}$
end for
$\boldsymbol{\mathcal{T}_{AS}}$ ⟵ the top $\boldsymbol{n}$ dichotomies $\subseteq \boldsymbol{\mathcal{T}_A}$
Return $\boldsymbol{\mathcal{T}_{AS}}$
```

In order to illustrate the augmenting process, let's consider a step-by-step example. Let $\boldsymbol{\mathcal{C}}$ be a confusion matrix for a multi-class problem with five classes as we show in Table 6.1. Here, we already set the diagonal of $\boldsymbol{\mathcal{C}}$ to zero. This is required because we want to find the classes that are in confusion and the diagonal represents the correct classifications rather than the mis-classifications.

	$\boldsymbol{C_1}$	$\boldsymbol{C_2}$	$\boldsymbol{C_3}$	$\boldsymbol{C_4}$	$\boldsymbol{C_5}$
$\boldsymbol{C_1}$	**0**	0	22	13	2
$\boldsymbol{C_2}$	0	**0**	48	26	2
$\boldsymbol{C_3}$	22	48	**0**	37	31
$\boldsymbol{C_4}$	13	26	37	**0**	1
$\boldsymbol{C_5}$	2	2	31	1	**0**

Table 6.1: An example of a confusion matrix $\boldsymbol{\mathcal{C}}$.

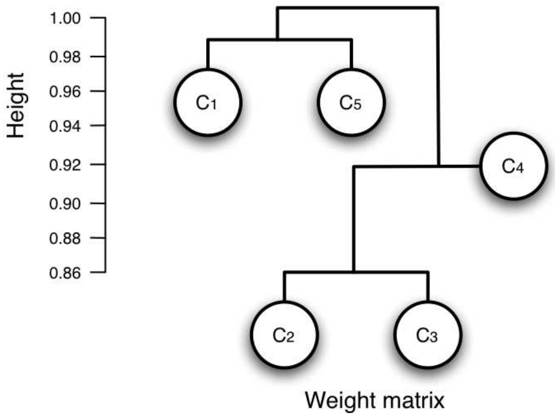


Figure 6.1: Hierarchical structure of $\boldsymbol{\mathcal{C}}$ based on the confusion relationship.

According to the Algorithm 8, in line 2, we perform the normalization of the confusion matrix $\boldsymbol{\mathcal{C}}$. After the normalization, high values represent low confusion. We shall present the reason of such normalization shortly. In line 3, we find a hierarchical representation of the confusions using a simple hierarchical clustering algorithm such as Agnes [78]. We show the resulting hierarchy of confusions in Figure 6.1. In the figure, we see that classes $\boldsymbol{C_2}$ and $\boldsymbol{C_3}$ are merged at the height (normalized confusion) of 0.86. This is consistent with the information in confusion matrix $\boldsymbol{\mathcal{C}}$ of Table 6.1, where $\boldsymbol{C_2}$ and $\boldsymbol{C_3}$ have 48 confusions in both directions. The classes $\boldsymbol{C_2}$, $\boldsymbol{C_3}$, and $\boldsymbol{C_4}$ are also in high confusion as we see in Table 6.1. On the other hand, the classes $\boldsymbol{C_1}$ and $\boldsymbol{C_5}$ has a common cluster with low confusion. Finally, the set $\boldsymbol{\{C_1 \ldots C_5\}}$ represents a confusion comprising all the classes.

In line 4 of the algorithm, we sort the sets according to the sum of the confusions normalized by the number of edges connecting the elements of the set excluding the self-references (e.g., $\boldsymbol{C_1}$ and $\boldsymbol{C_1}$). For instance, the confused set $\boldsymbol{\{C_2, C_3\}}$ has sum of $\boldsymbol{96 = 2 \times 48}$. As we have two entries excluding the self-references, the normalized weight of this set is 48. On the other hand, the set $\boldsymbol{\{C_2, C_3, C_4\}}$ has a sum of $\boldsymbol{222 \ = \ 2 \times 48 \ + \ 2 \times 37 \ + \ 2 \times 26}$. Given that this sum has six elements excluding the self-references it results a normalized weight of $\boldsymbol{37 = \frac{222}{6}}$. In the same way, $\boldsymbol{\{C_1 \ldots C_5\}}$ is the third set, in order, given that it has a sum of $\boldsymbol{364}$. As this sum has 20 elements excluding the self-references, the normalized weight of this set is $\boldsymbol{18.2 \ = \ 364 \ / \ 20}$.

In order to find the dichotomies representing each set of classes in confusion, we need to find a minimum cut in the confusion matrix representing this set of classes. This is the reason we normalized the confusion matrix $\boldsymbol{\mathcal{C}}$ in order to have low values representing higher confusions.

Consider the first set $\boldsymbol{\{C_2, C_3\}}$. This is the most trivial case to find the cut given that it has only two elements. Therefore, the dichotomy that represents this set can be $\boldsymbol{\vec{d}_{g1} = [0, 1, -1, 0, 0]^T}$ or its complement. However, the set $\boldsymbol{\{C_2, C_3, C_4\}}$ has more than two elements, and we need to find a way to partition this set and define its representative dichotomy. For that, we employ normalized cuts [165] in the sub-matrix representing this set. In this case, we find the best cut to be between $\{\boldsymbol{C_2}, \boldsymbol{C_4}\}$ and $\{\boldsymbol{C_3}\}$. Therefore, the dichotomy that represents this set can be $\vec{\boldsymbol{d}}_{\boldsymbol{g2}} = [\boldsymbol{0}, \boldsymbol{1}, -\boldsymbol{1}, \boldsymbol{1}, \boldsymbol{0}]^{\boldsymbol{T}}$ or its complement. Recall that such dichotomy means that we are interested in solving the confusion among classes $\boldsymbol{C_2}$, $\boldsymbol{C_3}$, and $\boldsymbol{C_4}$, designing a binary classifier specialized in separating elements from the classes $\boldsymbol{C_2}$, and $\boldsymbol{C_4}$ from elements of the class $\boldsymbol{C_3}$ and disregarding the rest.

6.4 Experiments and Results

In this section, we compare our *Affine-Bayes Multi-class* approach to: OVO, OVA, and ECOC approaches based on distances decoding strategies. We also compare our approach to Passerini et al. [127], who have proposed a Bayesian treatment for decoding assuming independence among all binary classifiers. For our *Affine-Bayes Multi-class*, we present three different results: one for normal *Affine-Bayes* (AB), one for *Affine-Bayes* and one stage of *Shrinking* (ABS), and finally, one with the iterative *Shrinking-Augmenting* (AB-OPT).

We validate our approach using two scenarios. In the first scenario, we use data sets with a relative small number of classes ($\boldsymbol{N_c} < \boldsymbol{30}$). For that, we use several UCI[3], and one NIST[4] data sets. In the second scenario, we have considered three large-scale multi-class applications: one for the Corel Photo Gallery (Corel)[5] data set, one for the Australian Sign Language (Auslan)[6], and one for the Amsterdam Library of Objects (ALOI)[7]. Table 6.2 presents the main properties of each data set we have used in the validation. Recall that, $\boldsymbol{N_c}$ is the number of classes, $\boldsymbol{N_d}$ if the number of features, and $\boldsymbol{N}$ is the number of instances.

Data set	**Source**	$\mathbf{N_c}$	$\mathbf{N_d}$	**N**
Pendigits	UCI	10	16	10,992
Mnist digits	NIST	10	785	10,000
Vowel	UCI	11	10	990
Isolet	UCI	26	617	7,797
Letter-2	UCI	26	16	20,000
Auslan	Auslan	95	128	2,565
Corel	Corel	200	128	20,000
ALOI	ALOI	1,000	128	108,000

Table 6.2: Data sets' summary. For the Auslan data set, we perform a dimensionality reduction to 128 features.

In the ECOC-based experiments, we have selected 10 random coding matrices. For each coding matrix, we perform 5-fold cross validation. For each cross-validation fold, we perform a 5-fold cross validation on the training set to estimate the CPTs. In all experiments, we have

[3] `http://mlearn.ics.uci.edu/MLRepository.html`

[4] `http://yann.lecun.com/exdb/mnist/`

[5] `http://www.corel.com`

[6] `http://mlearn.ics.uci.edu/MLRepository.html`

[7] `http://www.science.uva.nl/~aloi/`

used both Linear Discriminant Analysis (LDA) and Support Vector Machines (SVMs) [16] as base learners (examples of a week and a strong classifiers). For the clustering stage in our *Affine-Bayes Multi-class* solution, we report results using Fisher and Poland technique, as we showed in Section 6.3.2.

6.4.1 Scenario 1 (10 to 26 classes)

In Figures 6.2–6.4, we compare *Affine-Bayes* (AB) to ECOC based on Hamming decoding (ECOC), One-vs-One (OVO), One-vs-All (OVA), and Passerini's approach (PASSERINI) [127]. We show the One-vs-All (OVA) as the baseline and not as a function of the number of used base-leaners. In this experiment, *Affine-Bayes* uses three different coding matrices: AB-ECOC (normal *Affine-Bayes*), ABS-ECOC (*Affine-Bayes* with one stage of *Shrinking*), and AB-ECOC-OPT (*Affine-Bayes* with the *Shrinking-Augmenting* iterations).

The use of conditional probabilities and affine groups on *Affine-Bayes* to decode the binary classifications and create a multi-class prediction boosts the performance of ECOC-based approaches. This is also true for other UCI data sets not shown here such as *abalone*, *covtype*, and *yeast*.

For multi-class instances with small number of classes (e.g., $\boldsymbol{N_c \leq 26}$), weak classifiers (e.g., LDA) benefits more from *Affine-Bayes* than strong ones (e.g., SVMs). This important result shows us that when we have a problem with many classes, it may be worth using weak classifiers (e.g., LDA) which often are considerably faster than strong ones (e.g., SVMs).

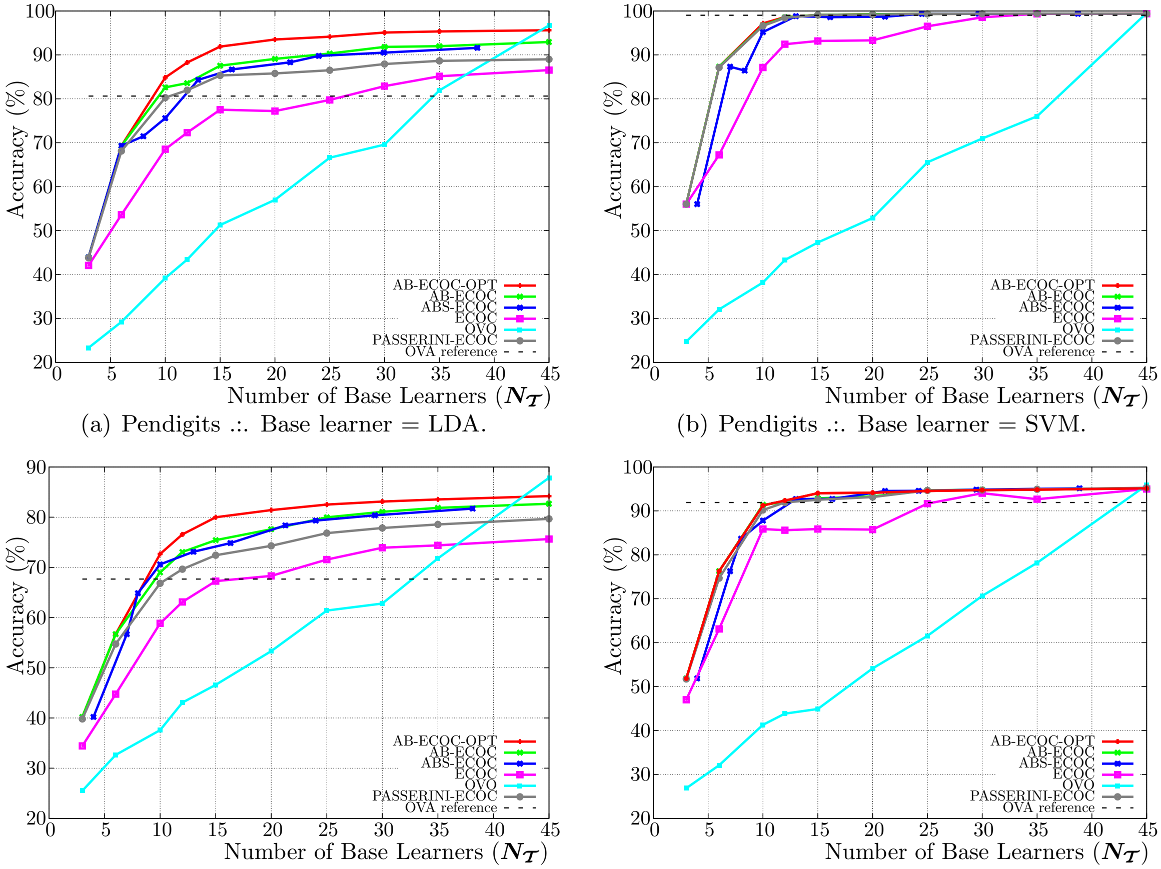
When possible, all One-vs-One dichotomies (OVO) produce better results. However, this approach implies in the use of all one-by-one dichotomies in the testing as well.

For the UCI and Nist small data sets, the *Affine Bayes* results are, in average, one standard deviation above Passerini's results when using SVM and, at least, two standard deviations above when using LDA. However, we have found that Passerini's assumption on independence for all dichotomies is not as robust as *Affine-Bayes* when the number of dichotomies and classes becomes larger (c.f., Sec. 6.4.2). This is also true for the all One-vs-One combinations. When the number of classes becomes larger, we have observed that these solutions becomes less discriminative. For small data sets, there is no much gain in using anything sophisticated.

This behavior is closely related to the curse of dimensionality, and most papers in the literature only show the performance going up to 30 classes which is not useful for large-scale problems. Here, we validate our approach for up to 1,000 classes.

Let's take a closer look at results in Figure 6.2. Here, *Affine-Bayes* with *Shrinking-Augmenting* option performs considerably well for Mnist and Pendigits data sets. For the SVM base learner, *Affine-Bayes* with *Shrinking-Augmenting* option is slightly better than its main competitor Passerini's solution. Clearly, the normal ECOC solution without the improvements of *Affine-Bayes* or Passerini is not as powerful. The most interesting lesson here is the cutoff we can use to obtain the same performance that we would get when using all One-vs-One base learners. Using 15 base learners in both data sets and both algorithms we already obtain very good results using *Affine-Bayes*. The second observation is that the One-vs-One combinations are only effective if they have all dichotomies at hand. It is not as powerful when some of them are missing.

For the experiments in this section, the average number of iterations for *Affine-Bayes* with the *Shrinking-Augmenting* option was 3.5 and the average number of dichotomies effectively replaced were 10% to 25%.



(a) Pendigits .:. Base learner = LDA.

(b) Pendigits .:. Base learner = SVM.

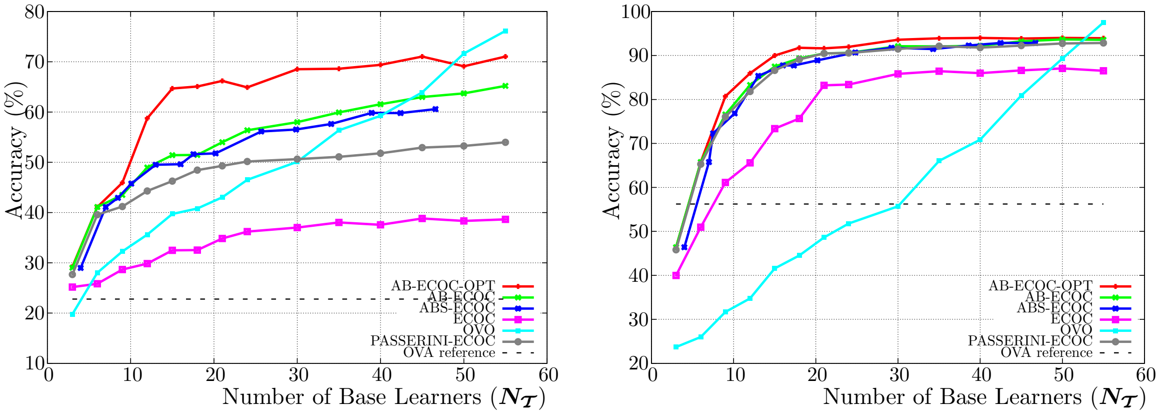
(c) Mnist .:. Base learner = LDA.

(d) Mnist .:. Base learner = SVM.

Figure 6.2: *Affine-Bayes* and derivatives (AB*) *vs.* ECOC *vs.* OVA *vs.* OVO *vs.* Passerini for Pendigits, and Mnist data sets considering LDA and SVM base learners.

We can draw similar conclusions from the experiments we show in Figure 6.3, for the Vowel data set. In this case, note how bad is the baseline approach of *One-vs-All*. *Affine-Bayes* with *Shrinking-Augmenting* option provides good performance as well as its version without such option. Here, we can use a cutoff of $\approx$ **20** base learners and still obtain good classification effectiveness.

In Figure 6.4, we present results for Isolet and Letter-2 data sets. All approaches based on *Affine-Bayes* present better performance than the Passerini, ECOC, and the baseline OVA solutions. Using LDA base learner, *Affine-Bayes* is, at least, five standard deviations more



(a) Vowel .:. Base learner = LDA.

(b) Vowel .:. Base learner = SVM.

Figure 6.3: *Affine-Bayes* and derivatives (AB*) *vs.* ECOC *vs.* OVA *vs.* OVO *vs.* Passerini for Vowel data set considering LDA and SVM base learners.

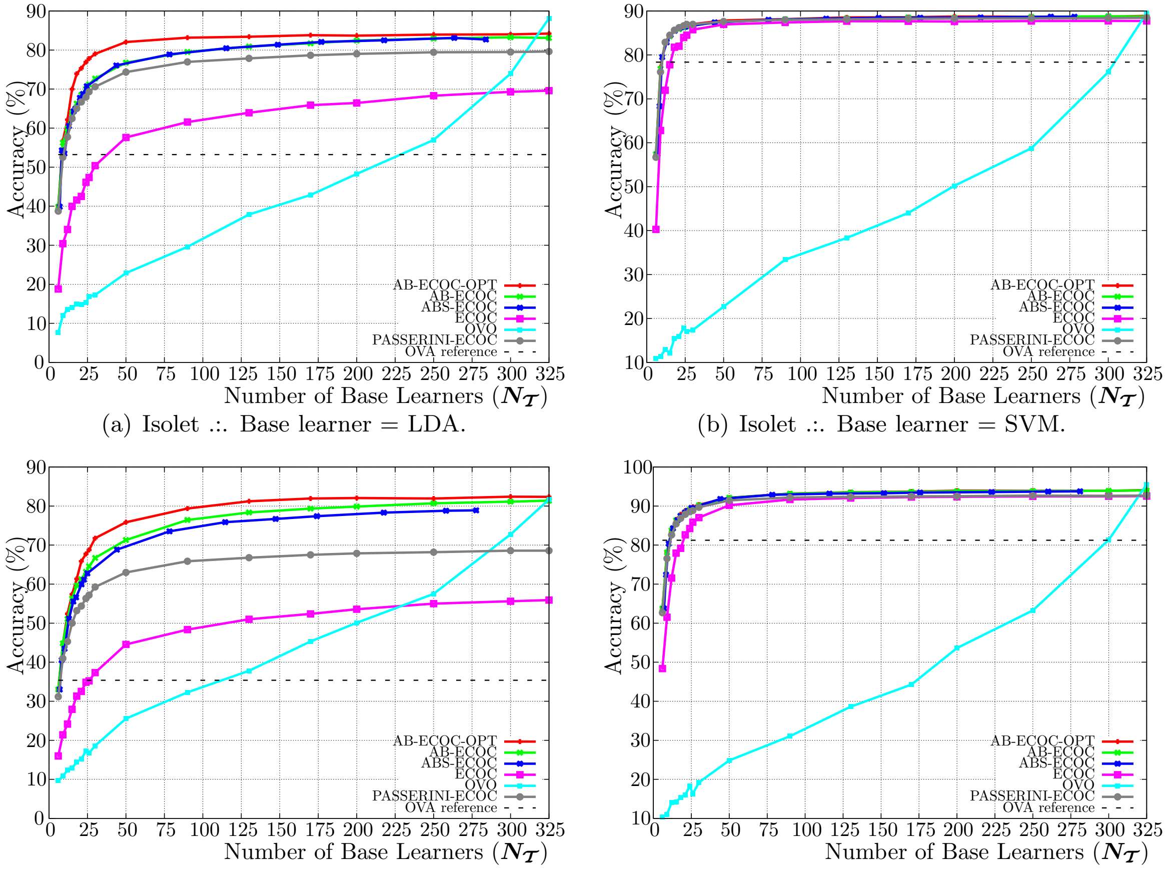
effective than Passerini's approch. Using SVM, this difference is about two standard deviations. In these experiments, we could use a cutoff of 50 base learners and still obtain acceptable results.

6.4.2 Scenario 2 (95 to 1,000 classes)

In this section, we consider three large-scale Vision applications: Auslan ($\boldsymbol{N_c} = \mathbf{95}$), Corel ($\boldsymbol{N_c} = \mathbf{200}$), and ALOI ($\boldsymbol{N_c} = \mathbf{1,000}$) categorization. In such applications, OVO is computationally expensive. For these scenarios, ECOC approaches with a few base learners seems to be more appropriate. In Figures 6.5–6.6, we show results using *Affine-Bayes* (AB-ECOC) *vs.* ECOC Hamming decoding and Passerini et al. [127] approaches for LDA and SVM classifiers.

Here, we emphasize the performance for a small number of base learners in comparison with the number of all possible separation choices. As we increase the number of classifiers, all approaches fare steadily better. As we show in the experiments, for scenarios with more than 30 classes, the independence restriction plays an important role and does not yield the best performance.

As the image descriptor is not our focus in this paper, we have used a simple extended color histogram with 128 dimensions [169] for Corel and ALOI data sets. Corel collection comprises broad-class images and it is more difficult to classify than the ALOI collection of controlled objects.



(a) Isolet .:. Base learner = LDA.

(b) Isolet .:. Base learner = SVM.

(c) Letter-2 .:. Base learner = LDA.

(d) Letter-2 .:. Base learner = SVM.

Figure 6.4: *Affine-Bayes* and derivatives (AB*) *vs.* ECOC *vs.* OVA *vs.* OVO *vs.* Passerini for Isolet and Letter-2 data sets considering LDA and SVM base learners.

Auslan

We show the experiments for Auslan data set in Figure 6.5. In this case, we provide the results for OVO and OVA as baselines. Auslan data set comprises $\boldsymbol{N_c = 95}$ classes. Therefore, OVO approach uses $\boldsymbol{\binom{95}{2} = 4,465}$ base learners.

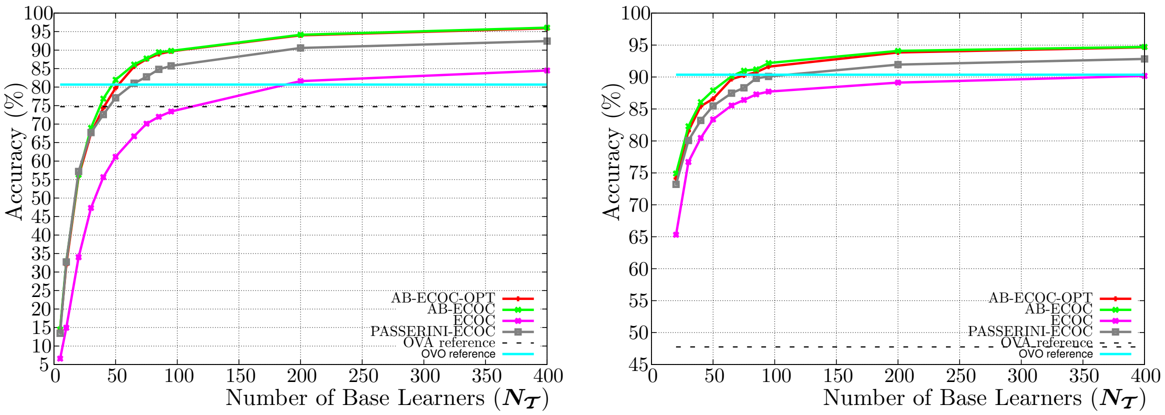
With LDA base learners, OVO with 4,465 dichotomies provides $\approx$ **80%** of accuracy while for 95 dichotomies, Passerini et al.'s approach results in $\approx$ **86%**, and *Affine-Bayes* solutions provide $\approx$ **90%** accuracy. As the maximum standard deviation (SD) across the cross-validation folds and different executions is $\approx$ **1%**, *Affine-Bayes* is four SDs more reliable than Passerini's solution.

With 400 dichotomies, or approximately 10% of all the One-vs-One combinations, *Affine-Bayes* yields $\approx$ **96%** accuracy for LDA base learner while Passerini et al.'s technique results in $\approx$ **92.4%** accuracy. The maximum SD here is $\approx$ **0.85%**.

With SVM base learners, OVO provides ≈ **90.3%** accuracy. With 95 dichotomies, Passerini's solution results in ≈ **90%** accuracy. On the other hand, *Affine-Bayes* results in ≈ **92.2%** accuracy. The maximum standard deviation here is **0.98%**. Therefore, using 95 dichotomies *Affine-Bayes* is ≈ **2** SDs above Passerini's solution and OVO.

With SVM and 400 dichotomies, or approximately 10% of all the One-vs-One combinations, *Affine-Bayes* provides ≈ **95%** accuracy or, at least, ≈ **2** SDs more effective than Passerini et al.'s solution and ≈ **5** SDs than OVO and the other solutions. Just for the sake of comparison, k-Nearest Neighbor[8] (k-NN, $\boldsymbol{k = 1}$) provides ≈ **77%** accuracy on this data set.

Finally, for this data set, *Affine-Bayes-OPT* is not statistically different than the normal *Affine-Bayes* regardless the base learner.



(a) Auslan .:. Base learner = LDA.

(b) Auslan .:. Base learner = SVM.

Figure 6.5: *Affine-Bayes* and derivatives (AB*) *vs.* ECOC *vs.* OVA *vs.* OVO *vs.* Passerini for Auslan data set considering LDA and SVM base learners.

Corel and ALOI

In this section, we provide results for Corel and ALOI data sets. In Figure 6.6, we show results for Corel collection. In this case, *Affine-Bayes* improves the effectiveness with respect to Passerini's and other approaches. For 200 dichotomies and base learner, *Affine-Bayes* results in **21%** accuracy. In spite of the reduction in the number of dichotomies, *Affine-Bayes* with one stage of *Shrinking* still provides good effectiveness with respect to the other solutions. Recall that Corel data set comprises broad-class images and it is more difficult to classify than the ALOI collection of controlled objects.

Figure 6.7 shows results for ALOI collection. When we use 200 dichotomies and LDA base learner in the training for ALOI data set, *Affine-Bayes* provides an average accuracy of 80% against 68% accuracy of Passerini's solution. In addition, for SVM base learner, with the

[8] Not shown in the plots.

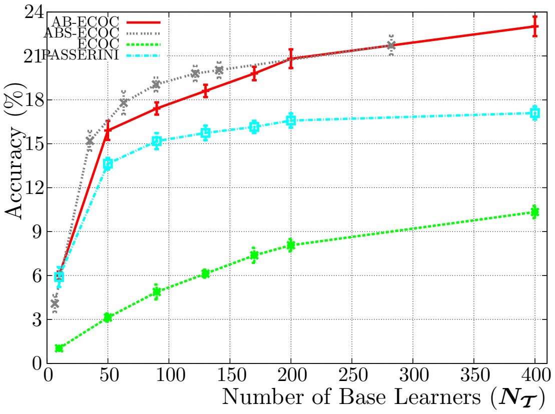
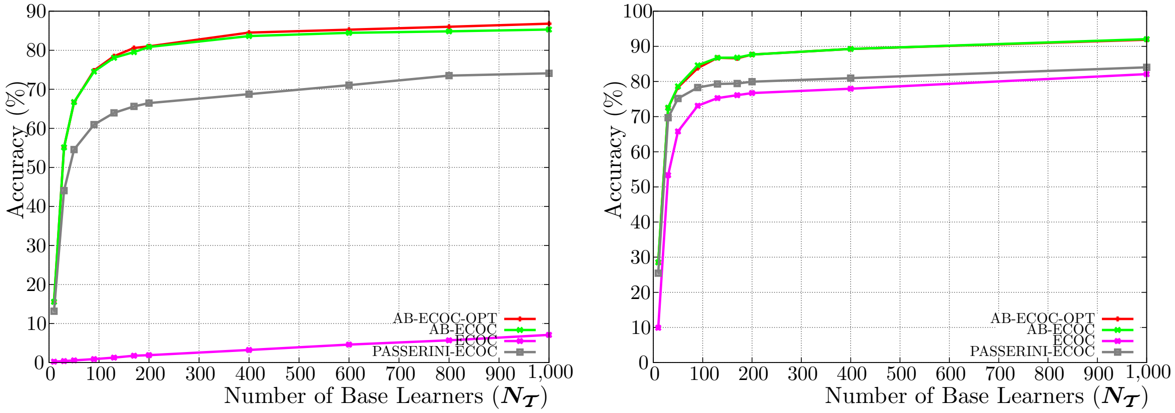


Figure 6.6: *Affine-Bayes* and derivatives (AB*) *vs.* ECOC *vs.* Passerini for Corel data set considering LDA base learner.

same 200 dichotomies, *Affine-Bayes* provides $\approx \mathbf{88\%}$ accuracy against $\approx \mathbf{80\%}$ of Passerini. The maximum standard deviation across the cross-validation folds and different executions is $\approx \mathbf{1.2\%}$. *Affine-Bayes* using 1,000 SVM base learners results in $\approx \mathbf{93\%}$ effectiveness against Passerini's $\approx \mathbf{84\%}$. Note that it was not viable to calculate the all One-vs-One accuracy here. It would require $\binom{\mathbf{1{,}000}}{\mathbf{2}} = \mathbf{499,500}$ base learners in the training and testing.



(a) ALOI .:. Base learner = LDA.

(b) ALOI .:. Base learner = SVM.

Figure 6.7: *Affine-Bayes* and derivatives (AB*) *vs.* ECOC *vs.* Passerini for ALOI data set considering LDA and SVM base learners.

6.5 Conclusions and Remarks

In this paper, we have addressed two key issues of multi-class classification: the choice of the coding matrix and the decoding strategy. For that, we have presented a new Bayesian treatment for the decoding strategy: *Affine-Bayes multi-class*.

We have introduced the concept of affine relations among binary classifiers and presented a principled way to find groups of high correlated base learners. Furthermore, we have presented a strategy to reduce the number of required dichotomies in the multi-class process and eliminate the less discriminative ones. In addition, we devised a strategy to automatically find new dichotomies and replace the ones tagged as less representatives in the *Shrinking* stage. We showed that we can use the two new procedures iteratively to complement the base *Affine-Bayes Multi-class* and boost the overall multi-class classification performance.

The advantages of our approach are: (1) it works independent of the number of selected dichotomies; (2) it can be associated with each one of the previous approaches such as OVO, OVA, ECOC, and their combinations; (3) it does not rely on the independence restriction among all dichotomies; (4) its implementation is simple and it uses only basic probability theory; (5) it is fast and does not impact the multi-class procedure.

Future work include the deployment of better policies to choose the initial coding matrix rather than random choice and the design of alternative ways to store the conditional probability tables other than sparse matrices and hashes.

Appendix

Table 6.3 presents some useful symbols we use throughout the text.

Acknowledgments

We thank FAPESP (Grants 05/58103-3, 07/52015-0, and 08/54443-2), and CNPq (Grants 309254/2007-8, 472402/2007-2, and 551007/2007-9) for the financial support.

$\boldsymbol{X}$	Data samples.
$\boldsymbol{x}$	An element of $\boldsymbol{X}$.
$\boldsymbol{Y}$	The class' labels of $\boldsymbol{X}$.
$\boldsymbol{y}$	An element of $\boldsymbol{Y}$.
$\boldsymbol{N}$	Number of elements of $\boldsymbol{X}$.
$\boldsymbol{N_c}$	Number of classes.
$\boldsymbol{N_d}$	The dimensionality X.
$\boldsymbol{C_L}$	The class labels.
$\boldsymbol{c}$	A class such that $\boldsymbol{c_i} \in \boldsymbol{C_L}$.
$\boldsymbol{\Omega}$	All possible dichotomies for $\boldsymbol{C}$.
$\boldsymbol{\mathcal{T}}$	A team of dichotomies such that $\boldsymbol{\mathcal{T}} \subset \boldsymbol{\Omega}$.
$\boldsymbol{d}$	A dichotomy such that $\boldsymbol{d} \in \boldsymbol{\mathcal{T}}$.
$\boldsymbol{N_{\mathcal{T}}}$	The number of dichotomies in $\boldsymbol{\mathcal{T}}$.
$\boldsymbol{M}$	A coding matrix
$\boldsymbol{\mathcal{O}_{\mathcal{T}}}$	A realization of $\boldsymbol{\mathcal{T}}$.
$\boldsymbol{\mathcal{A}}$	The affine matrix.
$\boldsymbol{\mathcal{G}_{\mathcal{D}}}$	The groups of affine dichotomies.
$\boldsymbol{g_i}$	Group of affine dichotomies such that $\boldsymbol{g_i} \subset \boldsymbol{\mathcal{G}_{\mathcal{D}}}$.
$\boldsymbol{\mathcal{C}}$	A confusion matrix.
$\boldsymbol{H}$	A hierarchical clustering representation of a confusion matrix $\boldsymbol{\mathcal{C}}$.

Table 6.3: List of useful symbols.

Capítulo 7

Conclusões

Nesta tese de doutorado, organizada na forma de coletânea de artigos, utilizamos várias técnicas de aprendizado de máquina e de classificação para extrair informações relevantes a partir de conjuntos de dados.

Mostramos que essas informações são valiosas e podem ser utilizadas para resolver diversos problemas em Processamento de Imagens e Visão Computacional. Particularmente, mostramos interesse em: categorização de imagens em duas ou mais classes, detecção de mensagens escondidas, distinção entre imagens digitalmente adulteradas e imagens naturais, autenticação, multi-classificação, entre outros.

7.1 Detecção de adulterações em imagens digitais

Com relação à análise forense de imagens, apresentamos um estudo comparativo e crítico das principais técnicas utilizadas atualmente. Mostramos que soluções para detecção de falsificações e mensagens escondidas em imagens ainda estão em sua infância. Discutimos também que a maior parte dessas soluções apontam para dois problemas relacionados ao aprendizado de máquina: a seleção das características a serem utilizadas no processo de classificação bem como as técnicas de classificação a serem empregadas.

Nos últimos anos temos visto uma crescente demanda por ferramentas para análise forense de imagens por diversas razões. Por um lado, segundo a perspectiva legal, essas técnicas podem ser essenciais para a investigação de muitos crimes, notadamente pornografia infantil. Sabemos que os crimes que utilizam imagens não se limitam a pornografia, vide o exemplo das atividades exercidas pelos cartéis colombianos comandados por Juan Carlos Abadía que tiravam vantagem da esteganografia para mascarar suas atividades ilegais. Por outro lado, segundo a perspectiva da comunidade de inteligência, a habilidade de analisar uma grande quantidade de dados para detecção de falsificações e conteúdo escondido é de interesse estratégico e de segurança nacional.

No entanto, os crimes óbvios não são necessariamente os mais perigosos. Diariamente, temos acesso a uma série de imagens com autenticidade duvidosa. A desinformação via meios de comunicação prevaleceu no último século, com imagens falsificadas sendo utilizadas constantemente como propaganda. Mas agora, com imagens sendo rotineiramente trabalhadas e utilizadas demagogicamente, é possível determinar sua autenticidade?

Em resposta a este desafio, apresentamos, nos Capítulos 2 e 3, várias considerações. Primeiramente, trabalhos a nível de decisão e fusão temporal de informações podem servir como uma excelente base para sistemas operacionais. A combinação de informação de vários algoritmos e técnicas pode nos trazer resultados mais confiáveis — especialmente quando não conhecemos exatamente o que estamos procurando. Segundo, o desenvolvimento de sistemas distribuídos com computação paralela pode exercer um importante papel para resolução de problemas relacionados à analise forense.

Finalmente, para a detecção de falsificações e de mensagens escondidas, a lição aprendida é que a comunidade tem agora um novo desafio: precisamos de algoritmos mais sofisticados para detectar os detalhes a respeito das manipulações encontradas nas imagens, não apenas o fato de que uma imagem foi manipulada. A despeito das limitações levantadas, o avanço do estado da arte nesta área continuará a melhorar nossa visão do desconhecido.

7.2 Randomização Progressiva

No Capítulo 4, apresentamos uma abordagem para meta-descrição de imagens denominada Randomização Progressiva (PR) para análise forense no contexto de detecção de mensagens escondidas e de classificação geral de imagens em categorias como *indoors*, *outdoors*, *geradas em computador* e *obras de arte*.

Como mostramos, a Randomização Progressiva é baseada em perturbações controladas dos *bits* menos significativos das imagens. Com tais perturbações, PR captura a separabilidade de algumas classes nos permitindo inferir algumas importantes informações a respeito das imagens analisadas.

A observação mais importante a respeito da Randomização Progressiva é que classes diferentes de imagens possuem comportamentos distintos quando submetidas a sucessivas perturbações. Por exemplo, um conjunto de imagens que não possui mensagens escondidas apresenta diferentes artefatos mediante sucessivas perturbações comparado a um conjunto de imagens que possui mensagens escondidas. Podemos fazer uma analogia com a compressão de arquivos. Ao comprimirmos um arquivo natural, sem nenhuma compressão prévia, temos um resultado. No entanto, ao comprimirmos um arquivo que já sofreu alguma compressão, o resultado dessa operação será diferente e, possivelmente, produzirá um arquivo maior que o arquivo de entrada.

Os bons resultados apresentados com essa técnica sugerem que ela, possivelmente, pode ser estendida também para outros cenários forenses. A técnica apresentada é capaz de detectar mensagens escondidas de tamanho médio (e.g., $\approx$ **25%** da capacidade) com qualidade superior a **90%**. Esse valor seria, por exemplo, o mínimo de informação que um indivíduo interessado em distribuir pornografia infantil iria alterar em uma imagem típica de papel de parede ($\mathbf{1280 \times 1024}$

pixels) para esconder uma imagem JPEG de tamanho aproximadamente 80 a 100 *kilobytes*.

7.3 Fusão multi-classe de características e classificadores

Ao estudarmos o problema de categorização multi-classe com imagens *indoors*, *outdoors*, *geradas em computador* e *obras de arte*, descobrimos que um problema de classificação multi-classe poderia ser mais bem resolvido a partir da combinação de diferentes características e classificadores. A principal lição desse capítulo é que a combinação de características e classificadores pode ser mais promissora do que o tradicional enfoque na busca de um descritor universal que resolva todo o problema.

Dado que cada classe tem certas particularidades, encontrar uma única característica geral que capture todas as propriedades é uma tarefa complexa. Embora a fusão de características seja bastante eficaz para alguns problemas, ela pode produzir resultados inesperados quando as diferentes características não estão normalizadas e preparadas de forma adequada. De forma geral, a combinação de várias características no mesmo vetor de descrição tende a requerer mais mais elementos para o treinamento devido à maldição da dimensionalidade. Adicionalmente, em certas ocasiões, alguns classificadores produzem melhores resultados para determinados descritores do que para outros, sugerindo que a fusão em nível de classificadores também poderia trazer bons resultados.

Nesse sentido, no Capítulo 5, nós desenvolvemos uma técnica para fusão de classificadores e características no cenário multi-classe através da combinação de classificadores binários. Nós definimos a binarização de classes como um mapeamento de um problema multi-classe para vários problemas binários (dividir para conquistar) e a subsequente combinação de seus resultados para derivar a predição multi-classe. Com essa técnica, podemos utilizar as características mais descritivas para determinadas configurações do problema bem como classificadores específicos para um conjunto de classes em confusão.

Nós validamos nossa abordagem numa aplicação real para classificação automática de frutas e legumes. Para essa aplicação, criamos um banco de dados com mais de 2600 imagens coletadas no centro de abastecimento de frutas e legumes de Campinas (CEASA). Esse banco de dados está disponível gratuitamente na *internet*[1].

7.4 Multi-classe a partir de classificadores binários

Outra questão importante que encontramos no decorrer desse trabalho diz respeito a como alguns poderosos classificadores binários (e.g., SVMs) podem ser estendidos para o cenário multi-classe de forma efetiva e eficaz. Sabemos que, como o SVM, vários outros classificadores foram originalmente desenvolvidos para classificação de problemas binários.

[1] `http://www.liv.ic.unicamp.br/~undersun/pub/communications.html`

Em tais casos, a abordagem mais comum é a de reduzir a complexidade do problema multi-classe para pequenos e mais simples problemas binários (dividir para conquistar).

Ao utilizar classificadores binários com algum critério final de combinação, muitas abordagens descritas na literatura partem do princípio de que os classificadores binários utilizados na classificação são independentes e aplicam um sistema de votação como política final de combinação. Entretanto, como discutimos no Capítulo 6, a hipótese da independência não é a melhor escolha em todos os casos.

Nesse sentido, no Capítulo 6, nós abordamos o problema de classificação multi-classe introduzindo o conceito de relações afins entre classificadores binários conhecidos também por dicotomias ou classificadores base. Denominamos a técnica de *Affine-Bayes*. A principal lição desse capítulo é que a combinação de pequenos classificadores pode ser bastante efetiva na resolução de um problema multi-classe.

Nós apresentamos uma forma efetiva de agrupar dicotomias altamente correlacionadas não supondo independência entre todas elas. Dentro de um grupo, há grande dependência entre os classificadores, enquanto que cada grupo é independente dos outros.

Introduzimos também uma estratégia para eliminar as dicotomias menos importantes no processo de multi-classificação e uma estratégia para desenvolver novas dicotomias para reposição daquelas menos eficazes. Mostramos também que essas duas estratégias podem ser utilizadas iterativamente para refinar os resultados do algoritmo base do *Affine-Bayes*.

Finalmente, nossos experimentos comprovam que a solução apresentada torna possível resolver problemas complexos tais como de 100 ou 1000 classes a partir da combinação de poucos, mas poderosos, classificadores binários.

Referências Bibliográficas

[1] Ismail Acbibas, Nasir Memon, and Bulent Sankur. Image steganalysis with binary similarity measures. In *Intl. Conference on Image Processing (ICIP)*, volume 3, pages 645–648, Rochester, USA, 2002. IEEE.

[2] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(11):1475–1490, November 2004.

[3] Naif Alajlan, Mohamed S. Kamela, and George Freeman. Multi-object image retrieval based on shape and topology. *Signal Processing: Image Communication*, 21(10):904–918, 2006.

[4] E. Allwein, R. Shapire, and Y. Singer. Reducing multi-class to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research (JMLR)*, 1(1):113–141, Jan 2000.

[5] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka. Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks (TNN)*, 6(1):117–124, January 1995.

[6] Ross Anderson and Fabien Petitcolas. On the limits of steganography. *Journal of Selected Areas in Communications (JSAC)*, 16(4):474–481, May 1998.

[7] V. Athisos, A. Stefan, Q. Yuan, and S. Sclaroff. Classmap: Efficient multiclass recognition via embeddings. In *Intl. Conference on Computer Vision (ICCV)*, 2007.

[8] Ismail Avcibas, Mehdi Kharrazi, Nasir Memon, and Bulent Sankur. Image steganalysis with binary similarity measures. *Journal on Applied Signal Processing*, 2005:2749–2757, May 2005.

[9] Ismail Avcibas, Nasir Memon, and Bulent Sankur. Steganalysis based on image quality metrics. In *Intl. Workshop on Multimedia and Signal Processing (MMSP)*, 2001.

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

[11] Sevinc Bayaram, Ismail Avcibas, Bulent Sankur, and Nasir Memon. Image manipulation detection. *Journal of Electronic Imaging (JEI)*, 15(4):1–17, October 2006.

[12] S. Bayram, H. Sencar, and N. Memon. Source camera identification based on CFA interpolation. In *Intl. Conference on Image Processing (ICIP)*, Genova, Italy, 2005. IEEE.

[13] S. Bayram, H.T. Sencar, and N. Memon. Improvements on source camera-model identiciation based on CFA interpolation. In *WG 11.9 Int. Conf. on Digital Forensics*, Orlando, USA, 2006. IFIP.

[14] A. Ben-Hur, H.T. Siegelmann, D. Horn, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research (JMLR)*, 2:125–137, Feb 2001.

[15] Alexander Berg, Tamara Berg, and Jitendra Malik. Shape matching and object recognition using low distortion correspondences. In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 26–33, 2005.

[16] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 1 edition, 2006.

[17] Rainer Bohme. Weighted stego-image steganalysis for jpeg covers. In *Intl. Workshop in Information Hiding (IHW)*, Santa Barbara, USA, 2008. Springer.

[18] R. M. Bolle, J. H. Connell, N. Haas, R. Mohan, and G. Taubin. Veggievision: A produce recognition system. In *Intl. Workshop on Applications of Computer Vision (WACV)*, pages 1–8, Sarasota, USA, 1996.

[19] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Scene classification via pLSA. In *European Conference on Computer Vision (ECCV)*. Springer, 2006.

[20] Robert W. Buccigrossi and Eero P. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing (TIP)*, 8(12):1688–1701, December 1999.

[21] Stephen Cass. Listening in. *IEEE Spectrum*, 40(4):32–37, April 2003.

[22] E. D. Castro and C. Morandi. Registration of translated and rotated images using finite fourier transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 9:700–703, May 1987.

[23] O. Celiktutan, I. Avcibas, B. Sankur, and N. Memon. Source cell-phone identification. In *Intl. Conference on Advanced Computing and Communication (ADCOM)*, Tamil Nadu, India, 2005. Computer Society of India.

[24] R. Chandramouli and K. P. Subbalakshmi. Current trends in steganalysis: a critical survey. In *Intl. Conf. on Control, Automation, Robotics and Vision*, pages 964–967, Kunming, China, 2004. IEEE.

[25] Mo Chen, Jessica Fridrich, Jan Lukas, and Miroslav Goljan. Imaging sensor noise as digital X-Ray for revealing forgeries. In *Intl. Workshop in Information Hiding (IHW)*, Saint Malo, France, 2007. Springer.

[26] X.C. Chen, Y.H. Wang, T.N. Tan, and L. Guo. Blind image steganalysis based on statistical analysis of empirical matrix. In *Intl. Conference on Pattern Recognition (ICPR)*, pages 1107–1110, Hong Kong, China, 2006. IAPR.

[27] Kai San Choi, Edmund Lam, and Kenneth Wong. Automatic source camera identification using the intrinsic lens radial distortion. *Optics Express*, 14(24):11551–11565, November 2006.

[28] P. Clark and R. Boswell. Rule induction with CN2: Some improvements. In *European Working Session on Learning (EWSL)*, pages 151–163, 1991.

[29] Brian Coe. *The Birth of Photography: The Story of the Formative Years, 1800-1900*. Book Sales, –, 1 edition, 1990.

[30] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(5):603–619, May 2002.

[31] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning (ML)*, 20(3):273–297, March 1995.

[32] Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, and Tom Kalker. *Digital Watermarking and Steganography*. Morgan Kauffman, Burlington, USA, 2 edition, 2008.

[33] K. Crammer and Y. Singer. On the learnability and design of output codes for multi-class problems. *Journal of Machine Learning Research (JMLR)*, 47(2–3):201–233, Mar 2002.

[34] Florin Cutzu, Riad Hammoud, and Alex Leykin. Distinguishing paintings from photographs. *Computer Vision and Image Understanding (CVIU)*, 100(3):249–273, March 2005.

[35] Anderson de Rezende Rocha. Randomização progressiva para esteganálise. Master thesis, Instituto de Computação, Universidade Estadual de Campinas (Unicamp), 2006.

[36] P. Debevec. Rendering synthetic objects into real scenes: bridging traditional and imge-based graphics with global illumination and high dynamic range photograph. In *ACM Siggraph*, pages 189–198, Orlando, US, 1998. ACM Press.

[37] Sintayehu Dehnie, Taha Sencar, and Nasir Memon. Identification of computer generated and digital camera images for digital image forensics. In *Intl. Conference on Image Processing (ICIP)*, Atlanta, USA, 2006. IEEE.

[38] T. G. Dietterich and G. Bakiri. Solving multi-class learning problems via error-correcting output codes. *Artificial Intelligence Research (JAIR)*, 2(1):263–286, January 1996.

[39] A. Dirik, H. Sencar, and N. Memon. Digital single lens reflex camera identification from traces of sensor dust. *IEEE Trans. On Inf. Forensics and Security*, 3(3):539–552, 2008.

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

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

[42] Hany Farid. Detecting hidden messages using higher-order statistical models. In *Intl. Conference on Image Processing (ICIP)*, volume 2, pages 905–908, Rochester, USA, Jun 2002. IEEE.

[43] Hany Farid. Creating and detecting doctored and virtual images: implications to the child pornography prevenction act. Technical Report TR 2004-518, Department of Computer Science - Dartmouth College, Hanover, USA, 2004.

[44] Hany Farid. Digital doctoring: How to tell the real from the fake. *Significance*, 3(4):162–166, 2006.

[45] Hany Farid. Exposing digital forgeries in scientific images. In *Multimedia and Security Workshop*, Geneva, Switzerland, 2006. ACM.

[46] Li Fei Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(4):594–611, 2006.

[47] I. Fisher and J. Poland. New methods for spectral clustering. Technical Report IDSIA-12-04, Dalle Molle Institute for Artificial Intelligence, Jun 2004.

[48] David Freedman, Robert Pisani, and Roger Purves. *Statistics.* W. W. Norton, 3 edition, 1978.

[49] Jessica Fridrich. Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes. In *Intl. Workshop in Information Hiding (IHW)*, pages 67–81, Toronto, Canada, 2004. Springer.

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

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

[52] Jessica Fridrich, D. Hogea, and M. Goljan. Steganalysis of jpeg images: Breaking the F5 algorithm. In *Intl. Workshop in Information Hiding (IHW)*, pages 310–323, Noordwijkerhout, Germany, 2002. Springer-Verlag.

[53] Jessica Fridrich, David Soukal, and Jan Lukas. Detection of copy-move forgery in digital images. In *Digital Forensic Research Workshop*, Cleveland, USA, 2003. DFRWS.

[54] Jiri Fridrich, Rui Du, and Meng Long. Steganalysis of LSB enconding in color images. In *Intl. Conference on Multimedia and Expo (ICME)*, volume 3, pages 1279–1282, 2000.

[55] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning.* Springer Verlag, 2001.

[56] D.D. Fu, Y.Q. Shi, D.K. Zou, and G.R. Xuan. JPEG steganalysis using empirical transition matrix in block dct domain. In *Intl. Workshop on Multimedia and Signal Processing (MMSP)*, pages 310–313, Victoria, Canada, 2006. IEEE.

[57] Z. Geradts, J. Bijhold, M. Kieft, K. Kurusawa, K. Kuroki, and N Saitoh. Methods for identification of images acquired with digital cameras. In *Enabling Technologies for Law Enforcement and Security*, volume 4232, –, 2001. SPIE.

[58] Thomas Gloe, Matthias Kirchner, Antje Winkler, and Rainer Bohme. Can we trust digital image forensics? In *ACM Multimedia (ACMMM)*, pages 78–86, Augsburg, Germany, 2007. ACM.

[59] Miroslav Goljan, Jessica Fridrich, and T. Holotyak. New blind steganalysis and its implications. In *Prof. of SPIE*, volume 6072, pages 1–13, 2006.

[60] Rafael Gonzalez and Richard Woods. *Digital Image Processing.* Prentice-Hall, 3 edition, 2007.

[61] Greg Goth. Steganography gets past the hype. *IEEE Distributed Systems Online*, 6(4):1–5, April 2005.

[62] K. Grauman and T. Darrell. Efficient Image Matching with Distributions of Local Invariant Features. In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 627–634, 2005.

[63] Steven Hand and Timothy Roscoe. Mnemosyne: Peer-to-peer steganographic storage. In *Intl. Workshop on Peer-to-Peer Systems*, volume 2429, pages 130–140, March 2002.

[64] Sarah V. Hart. Forensic examination of digital evidence: a guide for law enforcement. Technical Report NCJ 199408, National Institute of Justice NIJ-US, Washington DC, USA, September 2004.

[65] Jungfeng He, Zhouchen Lin, Lifeng Wang, and Xiaoou Tang. Detecting doctored JPEG images via DCT coefficient analysis. In *European Conference on Computer Vision (ECCV)*, pages 423–435, Graz, Austria, 2006. Springer.

[66] Gunther Heidemann. Unsupervised image categorization. *Image and Vision Computing (IVC)*, 23(10):861–876, October 2004.

[67] J. Huang, S. R. Kumar, M. Mitra, W-J Zhu, and R. Zabih. Spatial color indexing and applications. *Intl. Journal on Computer Vision (IJCV)*, 35(3):245–268, 1999.

[68] J. Sivic and B. Russell and A. Efros and A. Zisserman and and W. Freeman. Discovering objects and their location in images. In *Intl. Conference on Computer Vision (ICCV)*, pages 370–377, 2005.

[69] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems (NIPS)*, pages 487–493, 1998.

[70] Micah K. Johnson and Hany Farid. Exposing digital forgeries by detecting inconsistencies in lighting. In *ACM Multimedia and Security Workshop*, New York, USA, 2005. ACM.

[71] Micah K. Johnson and Hany Farid. Exposing digital forgeries in complex lighting environments. *IEEE Transactions on Information Forensics and Security (TIFS)*, 2(3):450–461, 2007.

[72] Micah K. Johnson and Hany Farid. Exposing digital forgeries through specular highlights on the eye. In *Intl. Workshop in Information Hiding (IHW)*, Saint Malo, France, 2007. Springer.

[73] Neil F. Johnson and Sushil Jajodia. Exploring steganography: Seeing the unseen. *IEEE Computer*, 31(2):26–34, February 1998.

[74] Neil F. Johnson and Sushil Jajodia. Steganalysis of images created using current steganography software. In *Intl. Workshop in Information Hiding (IHW)*, Portland, Oregon, 1998. Springer-Verlag.

[75] Frederic Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *Intl. Conference on Computer Vision (ICCV)*, volume 1, pages 604–610, 2005.

[76] K. Kuroki K. Kurosawa and N. Saitoh. CCD fingerprint method. In *Intl. Conference on Image Processing (ICIP)*, Kobe, Japan, 1999. IEEE.

[77] David Kahn. The history of steganography. In *Intl. Workshop in Information Hiding (IHW)*, pages 1–5, 1996.

[78] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, 1 edition, 1990.

[79] Andrew Ker. The ultimate steganalysis benchmark? In *Intl. Conference on Multimedia & Security (MM&Sec)*, pages 141–148, Dallas, USA, 2007. ACM.

[80] Andrew D. Ker. Optimally weighted least-squares steganalysis. In *Steganography and Watermarking of Multimedia Contents*, San Jose, USA, 2007. SPIE.

[81] Mehdi Kharrazi, Husrev Sencar, and Nasir Memon. Blind source camera identification. In *Intl. Conference on Image Processing (ICIP)*, Singapore, 2004. IEEE.

[82] A. Klautau, N. Jevtic, and A. Orlitsky. On nearest-neighbor ECOC with application to all-pairs multiclass support vector machines. *Journal of Machine Learning Research (JMLR)*, 4(1):1–15, Jan 2004.

[83] Jean Kumagai. Mission impossible? *IEEE Spectrum*, 40(4):26–31, April 2003.

[84] Yi Li and Linda G. Shapiro. Consistent line clusters for building recognition in cbir. In *Intl. Conference on Pattern Recognition (ICPR)*, volume 3, pages 30952–30957, 2002.

[85] Yue Li, Chang-Tsun Li, and Chia-Hung Wei. Protection of mammograms using blind steganography and watermarking. In *Intl. Symposium on Information Assurance and Security*, pages 496–500, August 2007.

[86] S. Lin, J. Gu, S. Yamazaki, and H. Y. Shum. Radimetric calibration from a single image. In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 938–945, Washington, USA, 2004. IEEE.

[87] Zhouchen Lin, Rongrong Wang, Xiaoou Tang, and Heung-Yeung Shum. Detecting doctored images using camera response normality and consistency. In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, USA, 2005. IEEE.

[88] Xuezheng Liu, Lei Zhang, Mingjing Lib, Hongjiang Zhang, and Dingxing Wang. Boosting image classification with lda-based feature combination for digital photograph management. *Pattern Recognition*, 38(6):887–901, June 2005.

[89] Y. Long and Y. Huang. Image based source camera identification using demosaicing. In *Intl. Workshop on Multimedia Signal Processing (MMSP)*, Victoria, Canada, 2006. IEEE.

[90] Roberto A. Lotufo and Alexandre Falcão. *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 18, chapter The ordered queue and the optimality of the watershed approaches. Kluwer Academic Publishers, 1 edition, June 2001.

[91] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Trans. On Inf. Forensics and Security*, 1(2):205–214, 2006.

[92] Jiebo Luo and Andreas Savakis. Indoor vs. outdoor classification of consumer photographs using low-level and semantic features. In *Intl. Conference on Image Processing (ICIP)*, pages 745–748. IEEE, 2001.

[93] S. Lyu and H. Farid. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1:111–119, 2006.

[94] Siwei Lyu. *Natural Image Statistics for Digital Image Forensics.* Phd thesis, Department of Computer Science - Dartmouth College, Hanover, USA, August 2005.

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

[96] Siwei Lyu and Hany Farid. Detecting hidden messages using higher-order statistics and support vector machines. In *Intl. Workshop in Information Hiding (IHW)*, pages 340–354, Dresden, Germany, 2002. Springer.

[97] Siwei Lyu and Hany Farid. Steganalysis using color wavelet statistics and one-class support vector machines. In *Symposium on Electronic Imaging*, San Jose, USA, 2004. SPIE.

[98] Siwei Lyu and Hany Farid. How realistic is photorealistic? *IEEE Transactions on Signal Processing (TSP)*, 53(2):845–850, 2005.

[99] Siwei Lyu, Daniel Rockmore, and Hany Farid. A digital technique for art authentication. *Proc. of the National Academy of Sciences (PNAS)*, 101(49):17006–17010, November 2004.

[100] M. Marszałek and C. Schmid. Spatial Weighting for Bag-of-Features. In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2118–2125, 2006.

[101] Mary Warner Marien. *Photography: A Cultural History.* Prentice Hall, –, 2 edition, 2006.

[102] Ueli Maurer. A universal statistical test for random bit generators. *Intl. Journal of Cryptology*, 5(2):89–105, February 1992.

[103] Columbia DVMM Research Lab. Columbia image splicing detection evaluation data set. Available at `http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm`, 2004.

[104] Errol Morris. Photography as a weapon. Available at `http://morris.blogs.nytimes.com/2008/08/11/photography-as-a-weapon/index.html`, The New York Times, July 11th, 2008.

[105] Folha de São Paulo. Para agência dos EUA, Abadía traficou no Brasil. Available at http://www1.folha.uol.com.br/fsp/cotidian/ff1003200801.htm (In Portuguese), March 10th, 2008.

[106] Herald Sun. Hello Kitty was drug lord's messenger. Available at `http://www.news.com.au/heraldsun/story/0,21985,23354813-5005961,00.html`, March 11th, 2008.

[107] Mike Nizza and Patrick Witty. In an iranian image, a missile too many. Available at `http://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many`, The New York Times, July 10^{th}, 2008.

[108] USPS. USPS – US Postal Inspection Service. Available at `www.usps.com/postalinspectors/ar01intr.pdf`, 2003.

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

[110] F. C. Mintzer, L. E. Boyle, A. N. Cases, B. S. Christian, S. C. Cox, F. P. Giordano, H. M. Gladney, J. C. Lee, M. L. Kelmanson, A. C. Lirani, K. A. Magerlein, A. M. B. Pavani, and F. Schiattarella. Toward online, worldwide access to vatican library materials. *IBM Journal of Research and Development*, 40(2):139–162, February 1996.

[111] M. Mirmehdi and M. Petrou. Segmentation of color texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(2):142–159, 2000.

[112] M. Moreira and E. Mayoraz. Improved pairwise coupling classification with correcting classifiers. In *European Conference on Machine Learning (ECML)*, 1998.

[113] Sheridan Morris. The future of netcrime now: Part 1 – threats and challenges. Technical Report 62/04, Home Office Crime and Policing Group, Washington DC, USA, 2004.

[114] A. Narasimhamrthy. Theoretical bounds of majority voting performance for a binary classification problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(12):1988–1995, Dec 2005.

[115] Anand Narasimhamurthy. Theoretical bounds of majority voting performance for a binary classification problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(12):1988–1995, December 2005.

[116] Ellias Nemer, Rafik Goubran, and Samy Mahmoud. Robust voice activity detection using higher-order statistics in the LPC residual domain. *IEEE Transactions on Speech and Audio Processing*, 9(3):217–231, March 2001.

[117] Tian-Tsong Ng and Shih-Fu Chang. Blind detection of photomontage using higher order statistics. In *Intl. Symposium on Circuits and Systems (ISCAS)*, pages 688–691, Vancouver, Canada, 2004. IEEE.

[118] Tian-Tsong Ng, Shih-Fu Chang, Ching-Yung Lin, and Qibin Sun. *Multimedia Security Technologies for Digital Rights Management*, chapter Passive-blind image Forensics, pages 1–30. Academic Press, Burlington, USA, 2006.

[119] Tian-Tsong Ng, Shih-Fu Chang, and Mao-Pei Tsui. Physics-motivated features for distinguishing photographic images and computer graphics. In *ACM Multimedia (ACMMM)*, pages 239–248, Singapore, 2005.

[120] NHTCU. National High Tech Crime Unit. www.nhtcu.org, 2008.

[121] Peter Nillius and Jan-Olof Eklundh. Automatic estimation of the projected light source direction. In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1076–1082, Hawaii, US, 2001. IEEE.

[122] Bruce Norman. *Secret warfare, the battle of Codes and Ciphers*. Acropolis Books, 1 edition, 1980.

[123] Aude Oliva and Antonio B. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Intl. Journal on Computer Vision (IJCV)*, 42(3):145–175, 2001.

[124] Nobuyuki Otsu. A threshold selection method from gray level histogram. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):66–66, January 1979.

[125] HweeHwa Pang, Kian-Lee Tan, and Xuan Zhou. StegFS: A steganographic file system. In *Intl. Conference on Data Engineering*, pages 657–667, March 2003.

[126] Greg Pass, Ramin Zabih, and Justin Miller. Comparing images using color coherence vectors. In *ACM Multimedia (ACMMM)*, pages 1–14, 1997.

[127] A. Passerini, M. Pontil, and P. Frasconi. New results on error correcting output codes of kernel machines. *IEEE Transactions on Neural Networks (TNN)*, 15(1):45–54, Jan 2004.

[128] Andrew Payne and Sameer Singh. Indoor vs. outdoor scene classification in digital photographs. *Pattern Recognition*, 38(10):1533–1545, 2005.

[129] Helen Pearson. Image manipulation: Csi: Cell biology. *Nature*, 434:952–953, April 2005.

[130] N. Pedrajas and D. Boyer. Improving multi-class pattern recognition by the combination of two strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(6):1001–1006, Jun 2006.

[131] Fabien Petitcolas, Ross Anderson, and Markus Kuhn. Information hiding - a survey. *Proc. of the IEEE*, 87(7):1062–1078, July 1999.

[132] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Attacks on copyright marking systems. In *Intl. Workshop in Information Hiding (IHW)*, pages 219–239, Portland, Oregon, 1998. Springer-Verlag.

[133] T. Pevny and J. Fridrich. Toward multi-class blind steganalyzer for jpeg images. In *Intl. Workshop on Digital Watermarking (IWDW)*, pages 39–53, Siena, Italy, 2005. Springer.

[134] Tomaz Pevny and Jessica Fridrich. Merging markov and dct features for multi-class jpeg steganalysis. In *Proc. of SPIE*, volume 6505, 2007.

[135] Birgit Pfitzmann. Information hiding terminology. In *Intl. Workshop in Information Hiding (IHW)*, volume 1174, pages 347–350, 1996.

[136] J. Platt, N. Christiani, and J. Taylor. Large margin DAGs for multi-class classification. In *Neural Information Processing Systems (NIPS)*, pages 547–553, 1999.

[137] Richard Popa. An analysis of steganographic techniques. Master thesis, Faculty of Automatics and Computers, The Politecnica University of Timisoara, 1998.

[138] Alin C. Popescu. *Statistical tools for digital image forensics*. PhD thesis, Department of Computer Science - Dartmouth College, Hanover, USA, December 2004.

[139] Alin C. Popescu and Hany Farid. Exposing digital forgeries by detecting duplicated image regions. Technical Report TR 2004-515, Department of Computer Science - Dartmouth College, Hanover, USA, 2004.

[140] Alin C. Popescu and Hany Farid. Statistical tools for digital forensics. In *Intl. Workshop in Information Hiding (IHW)*, Toronto, Canada, 2004. Springer.

[141] Alin C. Popescu and Hany Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing (TSP)*, 53(2):758–767, 2005.

[142] Niels Provos. Defending against statistical steganalysis. In *Usenix Security Symposium*, volume 10, pages 24–36, Washington, USA, 2001. Usenix.

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

[144] Niels Provos and Peter Honeyman. Hide and seek: an introduction to steganography. *IEEE Security & Privacy Magazine*, 1(3):32–44, March 2003.

[145] O. Pujol, P. Radeva, and J. Vitria. Discriminant ECOC: A heuristic method for application dependent design of ECOC. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(6):1007–1012, Jun 2006.

[146] Rajat Raina, Yirong Shen, Andrew Ng, and Andrew McCallum. Classification with hybrid generative/discriminative models. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[147] Anderson Rocha and Siome Goldenstein. Progressive randomization for steganalysis. In *Intl. Workshop on Multimedia and Signal Processing (MMSP)*, pages 314–319, Victoria, Canada, 2006. IEEE.

[148] Anderson Rocha and Siome Goldenstein. PR: More than Meets the Eye. In *Intl. Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

[149] Anderson Rocha and Siome Goldenstein. Steganography and steganalysis in digital multimedia: Hype or hallelujah? *Revista de Informática Teórica e Aplicada (RITA)*, 15(1):83–110, 2008.

[150] Anderson Rocha and Siome Goldenstein. Multi-class from binary: Divide-to-conquer. In *Intl. Conference on Computer Vision Theory and Applications*, pages –, Lisbon, Portugal, 2009. INSTICC.

[151] Anderson Rocha, Siome Goldenstein, Heitor A. X. Costa, and Lucas M. Chaves. Camaleão: um software de esteganografia para proteção e segurança digital. In *Simpósio de Segurança em Informática (SSI)*, 2004.

[152] Anderson Rocha, Siome Goldenstein, Heitor A. X. Costa, and Lucas M. Chaves. Segurança e privacidade na internet por esteganografia em imagens. In *Webmedia & LA-Web Joint Conference*, 2004.

[153] Anderson Rocha, Daniel C. Hauagge, Jacques Wainer, and Siome Goldenstein. Automatic produce classification from images using color, texture and appearance cues. In *Brazilian Symposium of Computer Graphics and Image Processing (SIBGRAPI)*, pages 3–10, Campo Grande, Brazil, 2008. IEEE.

[154] Anderson Rocha, Walter Scheirer, Siome Goldenstein, and Terrance E. Boult. The Unseen Challenge Data Sets. In *Intl. CVPR Workshop on Vision of the Unseen (WVU)*, pages 1–8, Anchorage, USA, 2008. IEEE.

[155] Benjamim Rodriguez and Gilbert Peterson. Steganalysis feature improvement using expectation maximization. In *Proc. of SPIE*, volume 6575, 2007.

[156] B.M. Rodriguez, G.L. Peterson, and S.S. Agaian. Steganography anomaly detection using simple one-class classification. In *Mobile Multimedia/Image Processing for Military and Security Applications*, pages 65790E.1–65790E.9, Orlando, USA, 2007. SPIE.

[157] Raul Rodriguez-Colin, Claudia Feregrino-Uribe, and Gershom Trinidad-Blas. Data hiding scheme for medical images. In *Intl. Conference on Electronics, Communications, and Computers (CONIELECOMP)*, pages 32–37, February 2007.

[158] Dario L. M. Sacchi, Franca Agnoli, and Elizabeth F. Loftus. Changing history: Doctored photographs affect memory for past public events. *Applied Cognitive Psychology*, 21(8):249–273, August 2007.

[159] Challa S. Sastry, Arun K. Pujari, and B. L. Deekshatulu. A fourier-radial descriptor for invariant feature extraction. *Intl. Journal of Wavelets, Multiresolution and Information Processing (IJWMIP)*, 4(1):197–212, 2006.

[160] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* Wiley & Sons, 2 edition, 1996.

[161] Taha Sencar and Nasir Memon. *Statistical Science and Interdisciplinary Research*, chapter Overview of State-of-the-art in Digital Image Forensics, pages –. World Scientific Press, Mountain View, USA, 2008.

[162] Navid Serrano, Andreas Savakis, and Jiebo Luo. A computationally efficient approach to indoor/outdoor scene classification. In *Intl. Conference on Pattern Recognition (ICPR)*, pages 146–149. IAPR, 2002.

[163] Biren Shah, Vijay Raghavan, Praveen Dhatric, and Xiaoquan Zhao. A cluster-based approach for efficient content-based image retrieval using a similarity-preserving space transformation method. *Journal of the American Society for Information Science and Technology (JASIST)*, 57(12):1694–1707, 2006.

[164] Toby Sharp. An implementation of key-based digital signal steganography. In *Intl. Workshop in Information Hiding (IHW)*, volume 2137, pages 13–26, 2001.

[165] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):888–905, August 2000.

[166] Li Shi, Sui Ai Fen, and Yang Yi Xian. A LSB steganography detection algorithm. In *Personal, Indoor and Mobile Radio Communications (PIMRC)*, volume 3, pages 2780–2783, Beijing, China, Sep 2003. IEEE.

[167] Yun Q. Shi, Chunhua Chen, and Wen Chen. A natural image model approach to splicing detection. In *ACM Multimedia and Security Workshop*, pages 51–62, Dallas, USA, 2007. ACM.

[168] Yun Q. Shi, Guorong Xuan, Dekun Zou, Jianjiong Gao, Chengyum Yang, Zhenping Zhang, Peiqi Chai, Wen Chen, and Chunhua Chen. Image steganalysis based on moments of characteristic functions and wavelet decomposition, prediction, error-image, and neural network. In *Intl. Conference on Multimedia and Expo (ICME)*, pages 268–272, Amsterdam, The Netherlands, 2005. IEEE.

[169] R. Stehling, M. Nascimento, and A. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *Intl. Conference on Information and Knowledge Management (CIKM)*, pages 102–109. ACM, 2002.

[170] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. *ACM Transactions on Graphics (ToG)*, 24(3):861–868, 2005.

[171] Yagiz Sutcu, Sevinc Bayaram, Husrev Sencar, and Nasir Memon. Improvements on sensor noise based source camera identification. In *Intl. Conference on Multimedia and Expo (ICME)*, Beijing, China, 2007. IEEE.

[172] A. Swaminathan, M. Wu, and K. Ray Liu. Non-instrusive forensics analysis of visual sensors using output images. In *Intl Conference on Image Processing (ICIP)*, Atlanta, USA, 2006. IEEE.

[173] T. Takenouchi and S. Ishii. Multi-class classification as a decoding problem. In *IEEE Intl. Symposium on Foundations of Computational Intelligence (FOCI)*, pages 470–475, 2007.

[174] The Compuserve Group. Specification of the GIF image format. Online, July 1990.

[175] The Electronic Frontier Foundation. The customer is always wrong: a user's guide to DRM in online music. Online, 2007.

[176] Min Tsai and Guan Wu. Using image features to identify camera sources. In *Intl. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, 2006. IEEE.

[177] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[178] Michael Unser. Sum and difference histograms for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 8(1):118–125, January 1986.

[179] Derek Upham. JSteg Shell. Online, 1999.

[180] P. P. Vaidyanathan. Quadrature mirror filter banks, m-band extensions and perfect reconstruction techniques. *IEEE Signal Processing Magazine*, 4(3):4–20, 1987.

[181] Aditya Vailaya, Anil Jain, and Hong Jiang Zhang. On image classification: city images vs. landscapes. *Pattern Recognition*, 31:1921–1935, 1998.

[182] Paul Viola and Michael Jones. Robust real-time face detection. *Intl. Journal on Computer Vision (IJCV)*, 57(2):137–154, February 2004.

[183] Julia Vogel and Bernt Schiele. A semantic typicality measure for natural scene categorization. In *DAGM Annual Pattern Recognition Symposium*, 2004.

[184] Paul Wallich. Getting the message. *IEEE Spectrum*, 40(4):38–40, April 2003.

[185] Tong Wang and Ji-Fu Zhang. A novel method of image categorization and retrieval based on the combination of visual and semantic features. In *Intl. Conference on Machine Learning and Cybernetics (ICMLC)*, pages 5279–5283, Guangzhou, China, 2005.

[186] Weihong Wang and Hany Farid. Exposing digital forgeries in interlaced and deinterlaced video. *IEEE Transactions on Information Forensics and Security (TIFS)*, 2:438–449, Mar 2007.

[187] Weihong Wang and Hany Farid. Exposing digital forgeries in video by detecting duplication. In *ACM Multimedia and Security Workshop*, Dallas, USA, 2007. ACM.

[188] Peter Wayner. *Disappearing Cryptography – Information Hiding: Steganography & Watermarking*. Morgan Kaufmann, 2 edition, 2002.

[189] Markus Weber. *Unsupervised learning of models for object recognition*. Phd thesis, Caltech, Pasadena, United States, May 2000.

[190] Andreas Westfeld. F5 — a steganographic algorithm high capacity despite better steganalysis. In *Intl. Workshop in Information Hiding (IHW)*, volume 2137, pages 289–302, Pittsburgh, US, 2001. Springer-Verlag.

[191] Andreas Westfeld and Andreas Pfitzmann. Attacks on steganographic systems. In *Intl. Workshop in Information Hiding (IHW)*, pages 61–76, Dresden, Germany, 1999. Springer.

[192] T. Windeatt and R. Ghaderi. Coding and decoding strategies for multi-class learning problems. *Information Fusion*, 4(1):11–21, Jan 2003.

[193] G.R. Xuan, J.J. Gao, Y.Q. Shi, and D.K. Zou. Image steganalysis based on statistical moments of wavelet subband histograms in dft domain. In *Intl. Workshop on Multimedia Signal Processing (MMSP)*, pages 1–4, Shanghai, China, 2005. IEEE.

[194] G.R. Xuan, Y.Q. Shi, C. Huang, D.D. Fu, X.M. Zhu, P.Q. Chai, and J.J. Gao. Steganalysis using high-dimensional features derived from co-occurrence matrix and class-wise non-principal components analysis (CNPCA). In *Intl. Workshop on Digital Watermarking (IWDW)*, pages 49–60, Jeju Island, South Korea, 2006. IEEE.

[195] C. Young, C. Yen, Yi Pao, and M. Nagurka. One-class-at-time removal sequence planning method for multi-class problems. *IEEE Transactions on Neural Networks (TNN)*, 17(6):1544–1549, Nov 2006.

[196] D. Zhang and G. Lu. Review of shape representation and description. *Pattern Recognition*, 37(1):1–19, 2004.

[197] Philip Zimmermann. *The Official PGP User's Guide*. MIT Press, 1 edition, 1995.